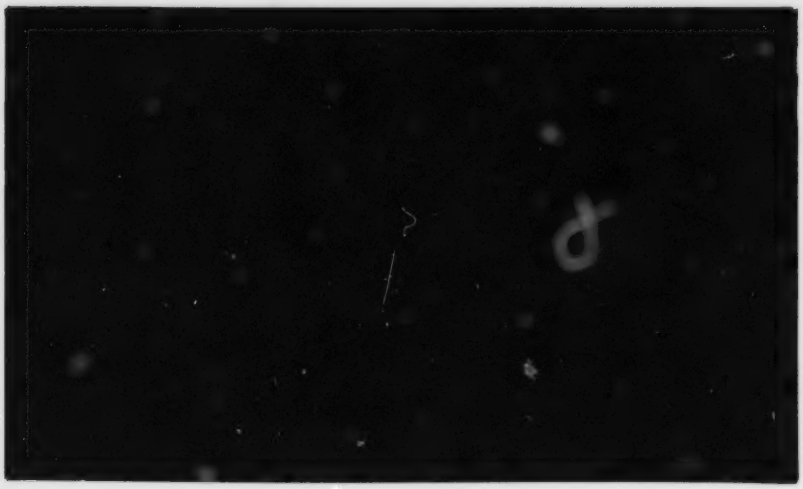
CENTRE
DE RECHERCHE
SUR LES TRANSPORTS
(CRT)
CENTRE
FOR RESEARCH
ON TRANSPORTATION

UNIVERSITÉ DE MONTRÉAL
ÉCOLE DES HAUTES ÉTUDES COMMERCIALES
ÉCOLE POLYTECHNIQUE



C.P. 6128, SUCCURSALE A
MONTRÉAL (QUÉBEC)
CANADA
H3C 3J7

TÉLÉPHONE : (514) 343-7575
TÉLEX : 05-24146 BIBPOLYTEC
TÉLÉCOPIEUR (FAX) : (514) 343-7121

CRT-640

STRATEGIC PLANNING OF FREIGHT TRANSPORTATION IN BRAZIL: METHODOLOGY AND APPLICATIONS

Part 3
THE STAN SYSTEM

Edited by
Michael Florian and Teodor G. Crainic

This research was supported by the International Development Research Center of Canada, Grant No. 3-P-82-1012 and in part by research grants NSERC (Canada), FCAR (Québec, Canada) and CNPq (Brazil).

Université de Montréal
Centre de recherche sur les transports - Publication # 640
Juillet 1989

90-01423/

Dépôt légal - 3e trimestre 1989

ABSTRACT

This is the final report of the cooperative project *Strategic Planning of Freight Transportation in Brazil*, carried out jointly by the Centre de Recherche sur les Transports of the Université de Montréal, the Pontifícia Universidade Católica do Rio de Janeiro and GEIPOT, the National Transportation Planning Company of the Brazilian Ministry of Transportation, with the financial support of the International Development Research Centre (Canada). The report consists of five parts:

Part I	Project Summary
Part II	Methodology: The New Contributions
Part III	The STAN System
Part IV	Specifying Data for the STAN System: Requirements, Concepts, Sources and Estimation Procedures
Part V	Applications.

The third part describes the main features of STAN, the interactive-graphic planning system for the strategic analysis of multimode multiproduct transportation networks at the regional and national levels, that was developped during the project.

KEY WORDS: Multimode Multiproduct Freight Transportation,
Network Optimization Models,
National/Regional Strategic Analysis and Planning of Freight Flows.
Interactive-Graphic System.

RESUME

Ceci est le rapport final du projet de coopération *Strategic Planning of Freight Transportation in Brazil*, réalisé conjointement par le Centre de recherche sur les transports de l'Université de Montréal, la Pontifícia Universidade Católica do Rio de Janeiro et GEIPOT, l'agence de planification nationale du ministère des transports du Brésil, avec le support financier du Centre de recherches pour le développement international (Canada). Le rapport a cinq parties:

Part I	Project Summary
Part II	Methodology: The New Contributions
Part III	The STAN System
Part IV	Specifying Data for the STAN System: Requirements, Concepts, Sources and Estimation Procedures
Part V	Applications.

La troisième partie décrit STAN, le système interactif-graphic pour l'analyse des réseaux de transport multimode multiproduit, au niveau régional ou national, développé à l'occasion du projet.

MOTS CLES: Transport multimode multiproduit de marchandises,
Modèles d'optimisation de réseaux,
Planification stratégique nationale/régionale,
Système interactif-graphique.

Table of Contents

I	**AN OVERVIEW OF THE STAN SYSTEM**	1
	1.1 GENERAL CONCEPTS THAT UNDERLY THE DESIGN OF STAN	2
	1.2 STAN - THE COMPUTER BASED SYSTEM	10
	1.3 BIBLIOGRAPHY	14
II	**BASIC SYSTEM CONCEPTS**	17
	2.1 INPUT AND OUTPUT	18
	2.2 NOTATION	37
	2.3 MODULES	44
	2.4 SOME IMPORTANT DIALOGS	54
	2.5 GRAPHIC OUTPUT	72

CHAPTER ONE

AN OVERVIEW OF THE STAN SYSTEM

STAN is an interactive-graphic multi-product multi-mode method for national or regional strategic planning of freight transportation. The strategic level of planning implies a medium to long term time horizon and a rather aggregate level of detail for the representation of the transport services provided. The contemplated alternatives represent usually anticipated (or predicted) changes in the origin to destination demands for the products considered and major changes to the transportation infrastructure. At the strategic level of planning, the aim is to obtain an overall estimation of the movements of freight of most (or all) products by most (or all) of the modes available: thus tactical decisions (such as when to schedule the trains that are operated) are not considered at this level.

A typical use of STAN is the comparison and evaluation of alternatives. Usually, an existing situation is described by one of the scenarios of the data base (base scenario). A simulation of the freight flows is carried out. Future situations are described by other scenarios which are constructed and contained in the data base. The simulation of freight flows are carried out on these scenarios as well. Then flows, link costs and delays, intermodal shipments and other performance indicators may be compared to those of the base scenario in order to carry out cost/benefit analyses.

STAN is the first interactive-graphic method which was designed for strategic planning of freight flows at the national or regional level. It allows the planner to engage in the planning process by using terminology that he is familiar with and obviates data processing type tasks. Once the data base is set up, the planner needs

not know or concern himself with the technical details of computer programming or computer systems. It also permits him the instantaneous visualization of input data, results of computations and information retrieved from the data base, all in graphic or list form. In the following, we describe the main features of this system.

Although STAN is specifically designed for freight transportation, it is based to a large extent on the design of EMME/2 (Babin et al. 1982, Spiess 1984), which is an interactive-graphic multi-mode method for urban and regional planning for the transportation of persons. In particular, the user interface is very similar to that used in EMME/2 (Babin et al.).

1.1 GENERAL CONCEPTS THAT UNDERLY THE DESIGN OF STAN

In STAN, the data that may be entered, modified and used for calculations falls into three main categories: networks, matrices and functions. Figure 1 below gives a schematic representation of the STAN data base, and is helpful in clarifying the descriptions that follow.

The **network** data may be entered, modified and displayed by using the Network Editor. The Matrix Editor performs the same operations for **matrices**, and provides as well facilities for matrix calculations and matrix balancing. All the relevant **function classes** and **functions** are entered and evaluated by the Function Editor. The STAN data base may also contain **demarcation lines**, **utility data** and a **log book**, which we refer to as the **auxiliary data**. Demarcation Lines may be superimposed on graphic displays to make them more readable. Utility data contain titles, module parameters, predefined windows,etc. A Log Book keeps

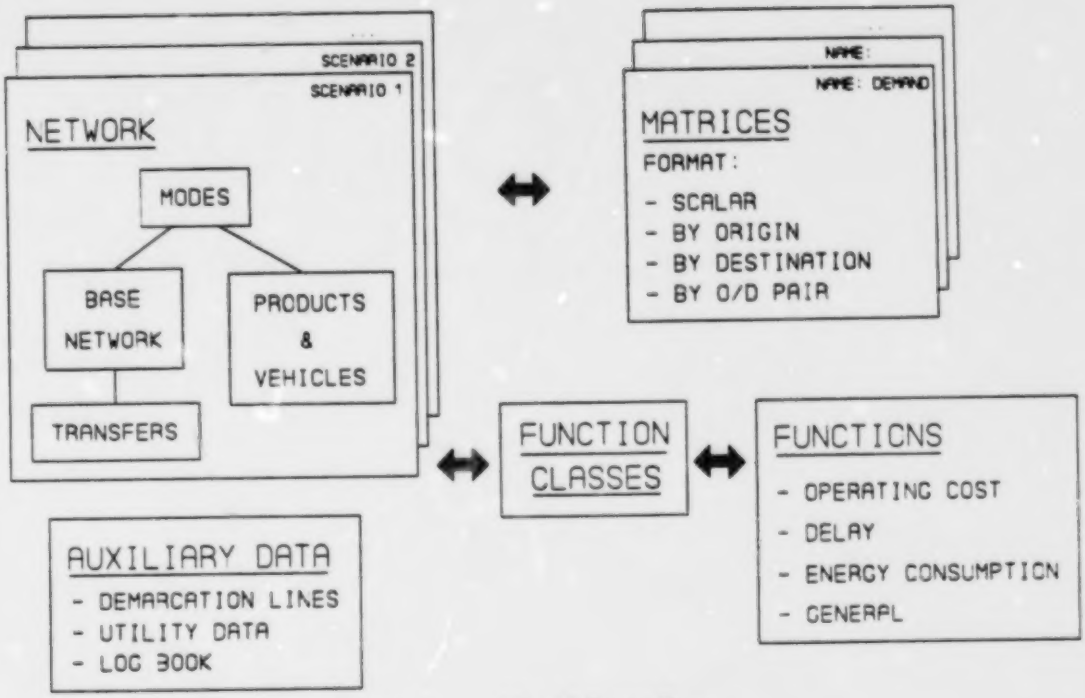


Figure 1 STAN Data Base

track of the use made of the various STAN modules.

Brief descriptions of the **Networks**, **Matrices** and **Functions** that may be entered and manipulated by STAN are given in the following sections.

1.1.1 The Networks

The transportation infrastructure that spans the region studied is represented by a multimodal network. The flows that can occur over this transportation network are described by specifying the products and the modes used for their transport. A **network scenario** is described by a set of **modes**, **products**, **vehicles**, **base**

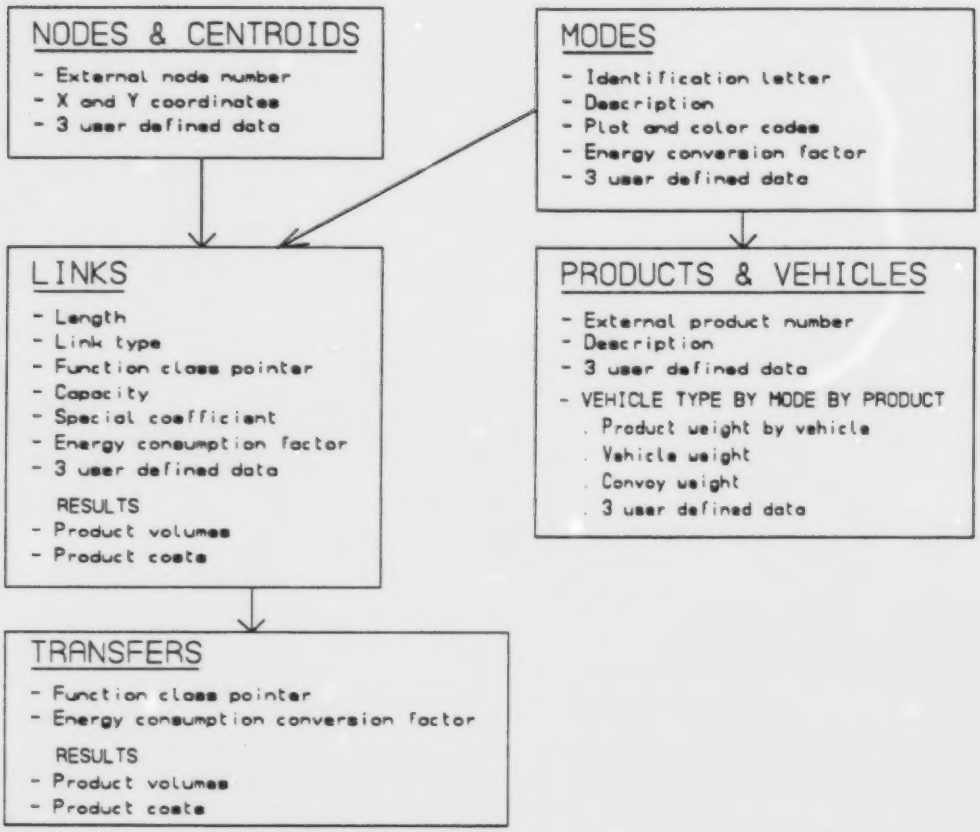


Figure 2 Network Scenario Hierarchy

network and **transfers**. Any of this data may be modified, at any time, provided that the hierarchy of the data structure is respected, as shown in Figure 2.

The **base network** is defined by the list of all the nodes and all the links between those nodes. In STAN parallel links are allowed between two nodes with the convention that a single mode is allowed on a given link. A path may contain successive links of different modes only if transfers are specified between them. A **transfer** is defined at a node, as a sequence from the mode of the incoming link, via the node, to the mode of the outgoing link. Thus, **transfers** represent intermodal

transhipments at nodes.

For each product, a vehicle type is specified for each mode. A **vehicle** is described by its weight, by the weight it can transport and by an optional convoy weight (used, for instance, to specify train length). The number of vehicle types equals the number of products multiplied by the number of permitted modes.

The user may specify up to three additional data elements for each **mode**, **product**, **vehicle**, **node** and **link**. Such **user data** may be observed flows or any associated data that the planner intends to use in subsequent analyses.

Each complete network data set (modes, products, vehicles, base network, transfers) makes up a **scenario**. The base period is one of these scenarios. The user may define a new scenario, by duplicating the data present in an existing scenario and by making the appropriate changes in any one of the network data components. All these manipulations may be performed interactively, graphically when appropriate. In STAN, the network data that corresponds to a **base period** is not conceptually different from data of contemplated scenarios.

1.1.2 The Matrices

The **matrices** that are handled in STAN may be full matrices, origin or destination vectors or scalars. These may contain various data related to the zone subdivision of the area studied, such as origin/destination demand matrices, production by origin and attraction by destination. A matrix may be both an input to or an output from a computational procedure.

The matrix editor, which is the same as that of EMME/2, has been designed to permit the user to choose a matrix format that avoids duplication of data in the data base. For instance, if a full matrix has the same value in all its cells, then the user may specify only a scalar matrix as its value. The matrix editor will then use only the space required to keep the scalar, until a full matrix is requested in a STAN module. Then it will automatically expand the scalar value to a full matrix with that value in all of its cells for the execution of that module. The same concept applies to given values for origins or destinations.

The matrix editor does not define any matrices by default. The definition of a matrix and its contents are left to the user. Thus, if an assignment routine expects the availability of origin/destination matrices, the user defines these matrices and ensures that they contain the correct data. This provides great flexibility in evaluating a given scenario with different O/D matrices.

Any network scenario created by the Network Editor may be used with any of the matrices present in the data base. The user identifies the relevant matrices that he wishes to use with a given scenario. The Matrix Editor of STAN also contains matrix balancing procedures in two and three dimensions. Their judicious use, with the appropriate matrices, permits the user to scale matrices or to implement some versions of classical spatial interaction (trip distribution) models.

Whenever the matrix editor expects a zone number, a zone group can be specified, indicating that the operation applies to all zones within the group. A zone group is a set of zones. A set of zone groups that includes all the zones is called an ensemble. Within an ensemble, a zone may be in only one group. The user of STAN may specify up to 26 ensembles, each containing up to 100 zone groups. The advantage of using zone groups is that selection of subsets of zones may be

accomplished efficiently, which is useful in many applications.

The Matrix Editor includes a module which permits the user to perform matrix calculations. The user specifies the operations that he wants to perform as an algebraic expression. Such algebraic expressions may be stated by choosing from 16 logical and algebraic operands and 9 intrinsic functions. The variables that may be used in these expressions are the matrices of the STAN data bank, centroid numbers and zone groups.

1.1.3 The Functions

The **functions** that are used in STAN are subdivised into three types: operating cost functions, time delay functions and energy consumption functions. These functions may be specified for links and for transfers.

These functions are not part of the code and have to be specified by the user as algebraic expressions. When required, a function is evaluated with data that corresponds to the variables specified by the user in the algebraic expression that defines it.

A **function class** is an index that associates to a link (transfer) a triplet of functions: an operating cost, a time delay and an energy consumption function. Function classes are an efficient and compact way to associate functions to links and transfers. Several links (transfers) may share the same function class.

In addition to the types of functions described above, the user is free to define and display functions that are not employed in any of the standard calculations of STAN.

1.1.4 Assignment Procedures

The most general **assignment procedure** provided by STAN is a multi-mode multiproduct assignment method which minimizes the total cost of shipping the products considered, from origins to destinations, via the permitted modes (see Chapter 3, part II).

For each product, a subset of modes, which are permitted for a given assignment, may be specified. Moreover, the O/D demand matrix of a product may be decomposed and a similar subset of modes may be defined for each resulting O/D matrix. This mechanism allows the introduction of predefined mode choices into the STAN system. It is always possible to assign all the products to all the modes, if desired.

The assignment does not permit explicit capacities on the network. Thus, when the services of certain modes are subject to capacity restrictions (e.g., rail, ports), these may be modeled as congestion and penalty functions, usually integrated into the delay cost functions.

The assignment procedure may be carried out by using delay functions, in units of time, cost functions in currency units, energy costs in the appropriated units or generalized costs (weighted combinations of time delays, operating costs and energy costs). The interaction between the costs of different products on different modes is considered explicitly in the assignment procedure of STAN.

Another computational algorithm that will be provided by the STAN system is a multiproduct multimode maximum flow method. This facility allows the evaluation of the maximal amounts of certain commodities that can be transported with the

existing transportation infrastructure and thus, may be very useful when considering major changes in the demand.

1.1.5 The Results

STAN permits the user to obtain a wide variety of **results** in interactive graphic form or as a printed output. The main feature of the results is the possibility of interactive comparison of scenarios with graphical displays. While the main results pertaining to comparison of scenarios are related to flows and costs, a wide variety of other results may be obtained using the user defined data.

Worthy of emphasis is that, unlike a "batch" code where each successful execution terminates with a particular set of results, an interactive graphic system allows the user to obtain results of different types during a particular session. The notion of "result" is thus different from that of a batch code and may be considered to consist of the entire gamut of displays, results of computational procedures, data bank queries and scenario comparisons.

1.1.6 Auxiliary Data

The Auxiliary Data consists of **demarcation lines**, **log book** and **utility data**.

The user may define **Demarcation Lines** that may be superimposed on a graphical output. A demarcation line may identify geographical characteristics of the area such as rivers or mountains, or certain regions of the country such as provinces, states, etc... .

A **Log Book** keeps record of the identity of the user and of the modules and elements of the data base used during all STAN sessions. The Log Book is not created automatically; this has to be done by the user.

Utility Data refers to certain data which are related to the use of STAN. These data may be transfered from one STAN data base to another to facilitate the construction of new data bases.

1.2 STAN - THE COMPUTER BASED SYSTEM

STAN has been coded in standard ANSI Fortran 77 and has been designed for easy transferability to various computer makes. It has been developed on a supermicrocomputer (MC68000), however it has been adapted for IBM PC/XT or AT equipped with the DSI-32 and DSI-780 co-processor cards, and the the SUN workstations. It may also be easily adapted for other computers using various operating systems.

The code of STAN has a modular structure. Each one of the modules (see Figure 3) is an independent program but all the modules share the STAN subroutine library. All data transfers between modules occur only via the data base. Figure 4 gives a schematic representation of the program structure.

```
1.  U T I L I T I E S
1.11 Direct data base manipulations
1.21 Log book consultation
1.22 Scenario manipulations
1.23 Input / modify / output utility data
1.31 Input          demarcation lines using batch entry
1.32 Input / modify demarcation lines interactively
1.34 Output         demarcation lines

2.  N E T W O R K  E D I T O R
2.01 Input / modify / output modes
2.02 Input / modify / output products
2.11 Input          base network using batch entry
2.12 Input / modify base network interactively
2.13 Plot           base network
2.14 Output         base network
2.21 Input          transfers using batch entry
2.22 Input / modify transfers interactively
2.23 Plot           transfers
2.24 Output         transfers
2.33 Plot shortest paths on network

3.  M A T R I X  E D I T O R
3.01 Input / modify / display zone groups
3.11 Input          matrices using batch entry
3.12 Input / modify matrices interactively
3.13 Plot           matrices
3.14 Output         matrices
3.15 Plot           matrix histograms
3.25 Matrix calculations
3.26 Matrix balancing

4.  F U N C T I O N  E D I T O R
4.01 Input / modify / display function classes
4.11 Input          functions using batch entry
4.12 Input / modify functions interactively
4.13 Plot           functions
4.14 Output         functions

5.  A S S I G N M E N T  P R O C E D U R E S
5.11 Prepare scenario for  assignment
5.21 Multiproduct multimode assignment
5.31 Maxflow calculations

6.  R E S U L T S
6.01 Result summary
6.13 Plot    product costs and volumes
6.14 List    product results
6.15 Plot    product vehicles and convoys
6.23 Compare product costs and volumes
6.33 Plot    shortest paths on network

9.  E N D  O F  S E S S I O N
```

Figure 3 The Modules Currently Making Up STAN

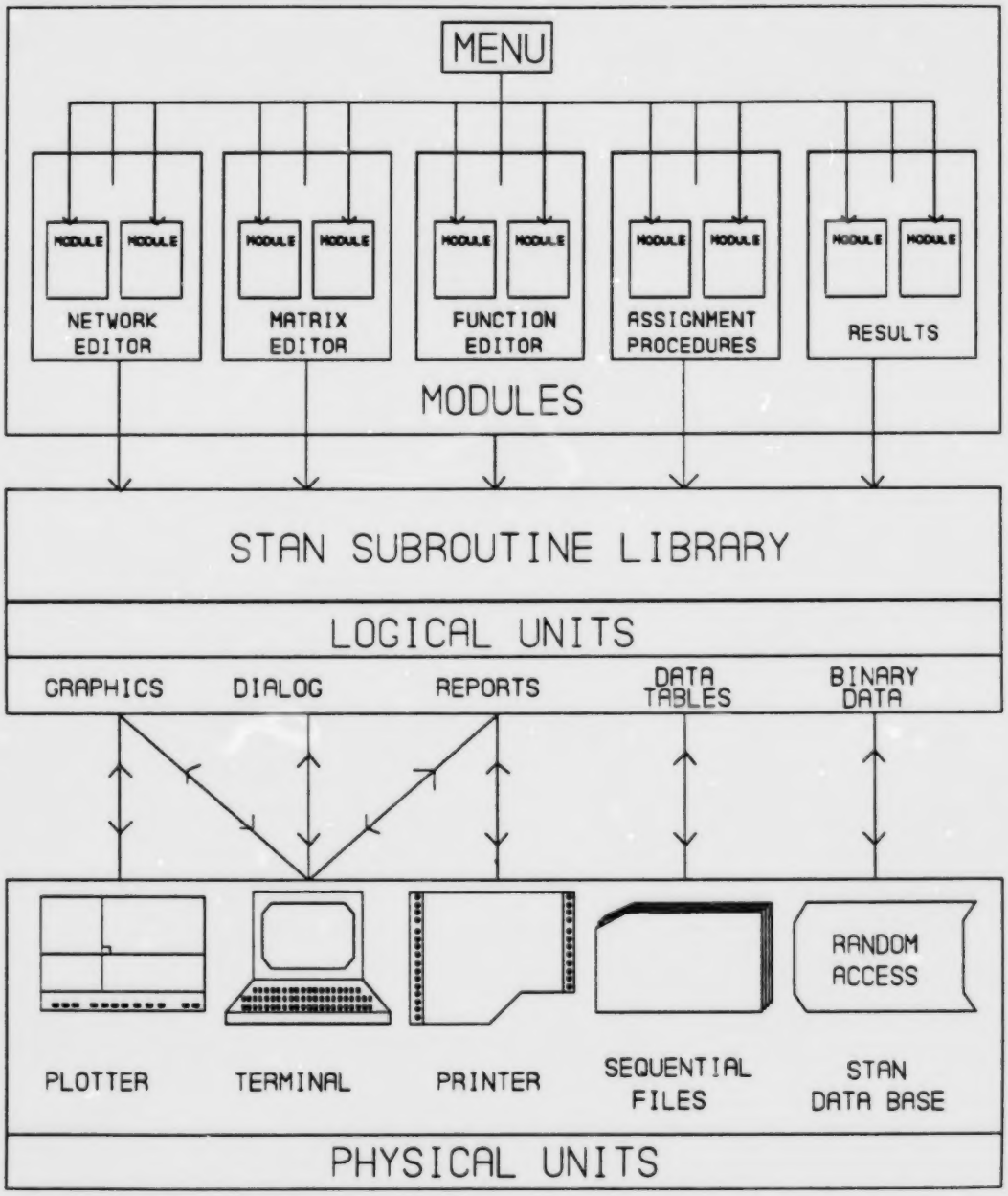


Figure 4 The Program Structure

At present, STAN is implemented for use on a wide variety of graphic terminals which include the Tektronix 4010, 4014, 4114, Visual 550 and all other Tektronix compatible monochrome terminals, as well as the Tektronix 4107, 4109, 4111, 4115, 4120, 4125 and the Seiko GR-1104 color graphic terminals. The same output may be drawn using Calcomp, Hewlett Packard, Houston Instruments or Tektronix digital plotters.

1.3 BIBLIOGRAPHY

Babin, A., Florian, M., James-Lefebvre, L. and H. Spiess (1982), *EMME/2: An Interactive Graphic Method for Road and Transit Planning*, **Transportation Research Record** 866, pp 1-9.

Constantin, I., Florian, M., James-Lefebvre, L., Larin, D. and H. Spiess, *EMME/2: User's Manual*, app. 600 pages, (frequent revisions).

Crainic, T.G., Florian, M. and J.E. Leal (1989), *A Model for the Strategic Planning of National Freight Transportation by Rail*, publication #518, Centre de recherche sur les transports, Université de Montréal, 1987, to appear in **Transportation Science**.

Crainic, T.G., Florian, M. and J. Guélat (1987), *O.R. Tools for Strategic Planning of National Freight Systems: Modeling*, **11th Triennial Conference on Operations Research,** IFORS, Buenos Aires, Argentine, 10-14 August.

Crainic, T.G., Florian, M. and J. Guélat (1987), *O.R. Tools for Strategic Planning of National Freight Systems: Algorithms*, **11th Triennial Conference on Operations Research,** IFORS, Buenos Aires, Argentine, 10-14 August.

Crainic, T.G. (1989), *An Interactive-Graphic System for the Analysis and Evaluation of Regional/National Transportation Networks*, **Transportation Networks and Regional Development**, U.S.S.R. Regional Science Association, May 23-26 1989, Leningrad, U.S.S.R., publication #632, Centre de recherche sur les transports, Université de Montréal.

Crainic, T.G., Florian, M. and J. Guélat, *STAN: User's Manual*, app. 500 pages, (frequent revisions).

Florian, M., Rico, N., and L.F.M. Vieira (1986), *A Multiproduct Max-Flow Algorithm*, TIMS/ORSA Joint National Meeting, Los Angeles.

Florian, M., Crainic, T.G. and J. Guélat (1987), STAN - *An Interactive-Graphic Method for Strategic Planning of Freight Flows*, **11th Triennial Conference on Operations Research,** IFORS, Buenos Aires, Argentine, 10-14 August.

Florian, M., Crainic, T.G. and J. Guélat (1987), STAN - *An Interactive-Graphic Method for Strategic Planning of Freight Flows*, publication #495, Centre de recherche sur les transports, Université de Montréal.

Guélat, J. (1986), *Une procédure de résolution du problème d'affectation multimode-multiproduit par décomposition sur un espace produit*, publication #488, Centre de recherche sur les transports, Université de Montréal.

Guélat, J., Florian, M. and Crainic, T.G. (1989), *A Multimode Multiproduct Network Assignment Model for Strategic Planning of Freight Flows*, publication #549, Centre de recherche sur les transports, Université de Montréal, 1987, to appear in **Transportation Science**.

Spiess, H. (1984), *Contributions à la théorie et aux outils de planification des réseaux de transport urbain,* Ph. D. thesis, département d'Informatique et de Recherche opérationnelle, publication #382, Centre de recherche sur les transports, Université de Montréal, 296p..

CHAPTER TWO

BASIC SYSTEM CONCEPTS

This chapter contains an outline of the general rules that a user should be familiar with before he starts to work with STAN. In order to make the system consistent and easy to use, some basic concepts were employed in its design that apply to the entire system. In the following, these basic concepts are described and illustrated with the details of their implementation; the general rules, that the user must assimilate in order to take advantage of the basic concepts, are given.

The chapter is organized in the following way: section 1 contains a description of the different logical categories of input/output (I/O) together with format considerations and hardware requirements; section 2 contains an introduction to the notation used for denoting and accessing different data elements; in section 3 the concept of the "menu" and the hierarchy of the modules is described; section 4 contains the details of some important sequences of dialog which occur frequently throughout the use of STAN, and, finally, section 5 contains descriptions of the standard graphic output and of the graphic commands which may be used to modify a given plot.

2.1 INPUT AND OUTPUT

2.1.1 Physical Devices and Logical Devices

Essentially, all the input and output, necessary for the use of a computer based system, originates from or is routed to some I/O peripheral device (such as the keyboard of a terminal for input or a line printer for output). STAN provides access to the following physical devices : terminal (sends and receives data, graphic or non-graphic), plotter, printer, sequential file storage device (such as a disk), error message output device and suppress file storage device (used when certain parts of the output should be suppressed). However, these physical devices are not addressed directly by the STAN programs; instead, the programs contain I/O instructions which write on or read from a logical device, which is defined by its logical content, rather than the physical device that it is allocated to. Each logical device is assigned to a physical device in a non permanent way, which permits changes of the assignments within the program, whenever such action is warranted during a session with STAN.

The table below lists all the logical devices that may be accessed by STAN and gives the physical devices that they are linked to by default:

- I/O terminal (terminal)
- background graphic device (plot file)
- data tables I/O (sequential disk files)
- print file (printer)
- error messages (error message disk file)

Even though STAN is meant to be used mainly in interactive mode, it may be preferable to use it in batch mode for certain repetitive or time consuming tasks. In that case, the terminal input file is connected to a sequential disk file, which con-

tains the anticipated answers to the interactive dialog, and the terminal output file is suppressed, since it would only contain the "question" part of the interactive dialog.

All the inputs and outputs of the STAN system fall into five categories: **dialog, reports, plots, data tables** and **error messages.** Each I/O category has its own standard formats and hardware requirements; during execution it is linked to one of the logical devices that it is designed for.

In sections 2.1.2 to 2.1.6 that follow, the standards and general rules that apply to each of these categories are described in detail.

2.1.2 The Dialog

The user interacts with the modules of STAN by means of a dialog, which consists of questions, answers and short descriptions. The output part of the dialog is generated by the system and consists of questions and short explanations; the input part of the dialog is the answer of the user to these questions.

There are three types of questions that are generated by the STAN system: **Yes/No** questions, **Select** questions and **Enter** questions. The rules for the dialog are as follows:

Yes/No question: Any question that is followed by a question mark (?) is a **Yes/No** question. The anticipated answer from the user is a **Yes** or a **No.** The abbreviations **y** and **n** are also accepted as valid answers. The user also has the choice of returning to the beginning of the module by answering **restart** or the abbreviation **r,** or to terminate the execution of the current module and return to the menu by answering **quit,** or the abbreviation **q.**

Any other answer to a **Yes/No** question is not accepted and the system, upon detecting an invalid answer, issues the message:

Answer must be: Yes, No, Restart, or Quit!.

The invalid answer is ignored and the user is expected to enter a valid answer. An example of dialog that involves a **Yes/No** question is:

Display node numbers? y

Select question: This type of question is generated by STAN whenever the user has to choose one of several alternatives. All **Select** questions are in the general format:

Select: <item to be selected>
1 =<alternative 1>
2 =<alternative 2>
. . .
n =<alternative n>
<answer>

The answer to a **Select** question is a number between 1 and n. The user may also choose to answer **restart (r)** or **quit (q)** , which return the user to the beginning of the module or to the menu. Any other answer to a **Select** question is not accepted, and the system issues the message:

A number between 1 and n (or q for quit, r for restart) must be entered!

The invalid answer is ignored and the user is expected to enter a valid answer. An example of a dialog that involves a **Select** question is:

Select: List device

1 = Terminal

2 = Printer

1

Enter question: This type of question is generated by STAN whenever data has to be entered by the user. There are two formats for the **Enter** question. The first format is

Enter: <Kind of data expected> = <*answer*>

and is employed only when an answer of one line is expected. The second format is employed when an answer of more than one line is expected, such as

Enter: <Kind of data expected>

= <*answer line 1*>

= <*answer line 2*>

...

= <*answer line n*>

= /

In this format, each time a line of answer is entered, the system generates an equal sign on the next line until the user terminates the input by entering a blank or a slash (/) or, sometimes, an asterix (*).

The answers to all **Enter** questions are given in free format for each line. The data is entered in the desired order and must be separated by a separator that is, one or several spaces (blank), a comma (,) or a colon (:). A slash (/) terminates the input record. In a repetitive **Enter** question, an asterix can be entered to specify all possible values for the item. Characters that follow a slash are ignored and may therefore be used as comments. The STAN free

format distinguishes between four types of data: integer values, real values, alphanumeric items and text. A text may contain separators but alphanumeric items may not. Both alphanumeric items and text may be delimited by quotes ("...") if the user desires it, but the system does not require it. If the data entered violates the rules of the STAN free format, the system issues the message:

Input error *xx* **at column** *cc*
Expected input: *t1 t2 t3 ....tn*

where ti indicates the type of the ith data item: I=integer, R=real, A=alphanumeric, T=text, K=keyword only. If the user replies with a blank or a slash, the data preceding the column in error will be processed. An example of a non repetitive (one line) **Enter** question is:

Enter: Matrix = mf21

An example of a repetitive **Enter** question is:

Enter: Link types (from, to)
=20,26
=30,36
=40,46
=/

In contrast to the **Yes/No** and **Select** questions, the **Enter** question does **not** permit the user to restart or quit prematurely.

2.1.3 Reports

All the list outputs that are produced by STAN in suitable formats are called reports. Such reports may be displayed on the terminal or may be sent to the printer. In either case, each page of the report has the standard STAN report header, which contains the project title and the relevant descriptions of scenario and/or matrices. When a report is sent to the printer, the header contains an additional line of text which gives the module number, the date, the installation name, the initials of the user and a page number.

An example is given below:

```
STAN  Module: 2.11        Date: 87 07 03          User: CRT_SUB3..ml          Page:    1
Project:      B R A Z I L  ---  S O U T H  E A S T  R E G I O N
Scenario 1000: R a i l ,  r o a d  a n d  p o r t  n e t w o r k s
-------------------------------------------------------------------------------------------
```

The page number is a count of all the report pages that have been sent to the printer since the creation of the data base, or since the page number has been reset to another desired value by using the Menu Command:

Page=<*new current page number*>

(See also section 2.3.1 of this chapter)

When STAN is used in batch mode, all reports are automatically sent to the printer. When STAN is used interactively, the following **Select** question must be answered immediately before the generation of a report:

Select: List device
1 = Terminal
2 = Printer

The user may thus choose the desired output device for each report. If the report is sent to the printer, the following message is displayed on the terminal:

***xxxx* Pages sent to print file**

If the report is displayed on the terminal, the system expects the user to input a space (blank) after each page is listed to indicate that he is ready for the next page. **Restart** (r) and **quit** (q) are also valid inputs at the end of a report page and return the user to the beginning of the module or to the menu.

The number of lines per page, for both the terminal and the printer, are device parameters; these parameters are defined in the device table (see section 2.1.7). Also defined in the device table is an end-of-page attention string which is displayed at the terminal after a page is completed and the system waits for a response. This attention string is up to 4 characters long and consists usually of a bell character (ASCII 7) followed by the characters '– –>'.

On some computer systems, the implementation of STAN allows for interrupts by the user. If this is the case, reports that are generated in interactive mode, but that are sent to the print file, may be interrupted prematurely. An interruption of a report only takes effect at the beginning of the next page and results in a message of the form

Report interrupted after *xxxx* pages.

2.1.4 Plots

All the graphic inputs and outputs that are made possible by STAN are called plots. The basic hardware required in order to use STAN in graphic mode are:

a) a plotting device which is capable of drawing a straight line between any two points of its plotting surface (such as a screen or a plotter);

b) a graphic input device, associated with the plotting device, which allows the identification of a point on the plotting surface (such as a cross hair cursor or a light pen).

Although the system does not require any minimal resolution of the graphic device, there are practical requirements imposed by the readability of the generated plots. A resolution of 500 points (horizontal) x 500 points (vertical) can be considered a minimal resolution required in order to yield graphic displays that are barely readable. A resolution of 3200 points x 4000 points (such as that offered by the Tektronix 4014 terminal with the Enhanced Graphics Module) yields graphic displays of very high quality.

Although STAN may be used with any graphic device that satisfies the above mentioned requirements, special care was given to make full use and take advantage of the special features of the Tektronix 4010 series of graphic terminals. If a terminal of this series is used with STAN, certain system parameters may be set to use the following features in order to make more efficient I/O graphics:

- four sizes of hardware generated characters;
- hardware generated dotted, dashed and dot-dashed lines;
- extended addressability;
- redundancy free address transmission.

STAN has also be adapted for high-resolution color-graphics terminals, such as the Tektronix 4115B and the Chromatics CGC 7900 series terminal. The colors are used to enhance the readability of the plots, which otherwise remain identical to those generated on monochrome terminals.

STAN distinguishes between two types of graphic devices: the foreground graphic device (normally the graphic terminal) and the background graphic device (normally a plot file). All plots may be generated on both types of graphic devices with the exception of graphic work sheets, a particular type of graphic I/O, which is restricted to the foreground graphic device. See also section 5 of this chapter, which describes in detail the various types of graphic I/O and their control commands.

2.1.5 Data Tables

Data tables are used in STAN when data are entered by using sequential card image files that are prepared outside the system. This way of entering data is called *input using batch entry*. Data tables may also be produced by the system from the contents of the data base. This way of producing data tables is called *punching a data table*.

2.1.6 Error Messages

The checking of input data in STAN is always done immediately after a data item has been entered. Most of the errors are therefore associated with a line of input. Once an error condition has been detected, STAN responds by the following actions:

a) A description of the error condition is sent to the corresponding output device, that is, to the terminal during a dialog or to the print device during the input of data tables. The error description usually allows the user to identify and correct the error. If the error occurs during a graphic operation in which no written error message may be sent to the screen, the error is indicated by three rings of the bell.

b) A standardized error message is sent to the error message device. This logical device is normally assigned to the terminal or the printer, which causes the standardized error messages to appear intermittently, as they occur, with all the other output. The standardized error message has the form:

Error (*m,t*) no= *n* mod= *x.xx* (yymmdd) p1 p2 p3

where *m* is the operation mode (I=interactive, B=batch), *t* is the type of error (F=fatal, W=warning) and *n* is the error number. Errors with numbers that are smaller than 1000 are generated by the module *x.xx* directly, whereas errors with numbers that are larger than 1000 are generated by one of the subroutines of the STAN library, that was called by the module *x.xx*. Along with the module number the date on which this module was compiled is indicated in the form *yymmdd*, which identifies the release version of the module. The parameters p1, p2 and p3 give details of the condition of the error and their main use is intended for program maintenance. Appendix A of the STAN Manual contains a complete list of all error messages.

c) If the error message was only a warning, the system resumes execution. If a fatal error occurs, execution is stopped if STAN is used in batch mode. If the fatal error occurs when STAN is used interactively, execution is resumed from the last question asked and the user may correct the error immediately with a valid answer.

2.1.7 Device Types and Device Table

STAN allows the user to use different types of equipment without having to reconfigure the programs. This is achieved by the definition of **device types,** each of which is associated with one or several of the devices: terminal, printer and plot file. The parameters associated with these devices are:

TERMINAL:

- Terminal initialization string (up to 32 chars)
- Terminal termination string (up to 32 chars)
- Dialog initialization string (up to 32 chars) and padding time
- Report initialization string (up to 16 chars) and padding time
- GIN initialization string (up to 16 chars)
- Plot initialization string (up to 32 chars) and padding time
- End-of-page attention string (up to 4 chars)
- Null character (ASCII code)
- Graphic device parameters (saved as a bit pattern describing characteristics of the graphic device)
- Number of different line patterns or colors
- Number of lines per report page
- Transmission speed (baud rate)

PRINTER:

- Report initialization string (up to 32 chars) and padding time
- Number of lines per report page

PLOT FILE:

- Plot initialization string (up to 32 chars) and padding time
- Null character (ASCII code)
- Graphic device parameters
- Number of line patterns or colors

For every device type the applicable devices and the values of the corresponding parameters are defined in the **device table**. This table is created whenever a new data base is built. Depending on the implementation the device table may be constant or may be read from an input file. If the device table is to be reinitialized in an already existing data base, switch 10 (see menu commands section 2.3.1) must be set to **on,** which causes the device table to be rewritten at the beginning of the next session. Note that switch 10 should normally be in the **off** position.

When a session is started, the user is first prompted to indicate the device types for terminal, printer and plot file he will be using during this session. For each device type the last device used is displayed and the user is asked whether he wishes a different device type. If yes, the available device types are displayed in the form of a **Select** question. If the device table entry for the selected terminal type contains no transmission speed, the user is asked to enter the baud rate (between 110 and 19200) he is currently using.

Example of such a dialog:

Current terminal type is: Non-graphic terminal

Change? yes

Select: Type of terminal

1= Tektronix 4014

2= Tektronix 4115B

3= Visual 550 (vt100 mode)

4= Seiko GR-1104 VT-100

5= Seiko GR-1104 VT-100 w. Tablet

6= ANSI terminal 24x80 non-graphic

7= PlotDev on IBM PC Color

8= PlotDev on IBM PC Monochrome

9= Non-graphic terminal

3

Enter: Baud rate=4800

Current printer type is: ASCII printer 60 lines/page

Change? yes

Select: Type of printer

1= ASCII printer 60 lines/page

2= ASCII printer 80 lines/page

2

Current plotfile type is: Tektronix 4014
Change?yes
Select: Type of plotfile
1= Tektronix 4014
2= Tektronix 4115B
3= Visual 550 (vt100 mode)
4= Seiko GR-1104 VT-100
5= Seiko GR-1104 VT-100 w. Tablet
6= gpl (PLOTFILE GENERIC (COLOR))
7= gpr (Tektronix 4014)
3

If some or all of the device types are known to remain unchanged for a STAN application, the corresponding parts of the above dialog may be suppressed by re-setting the switches 7 (terminal), 8 (printer) and 9 (plot file). The user may also change the current device types during a session by the use of the menu command **device** (see section 2.3.1). In this case the state of the switches 7 to 9 is irrelevant, all device types may be redefined.

The following table shows the parameters associated with a device type and their values for the Tektronix 4014 EGM graphic terminal. Note that the format used to define the device table is installation dependent. However, this need not concern the user, since the device table is normally provided at installation time and should only be changed by a STAN system programmer.

```
Tektronix 4014 EGM device table:

c   device name is:
    Tektronix 4014
c   device classes are:
    terminal    plotter
c   parameter switches effectively on are:
     0  1  3  4  5  6  7 19
    extrabyte= 1  arcsmooth= 1  charfont= 0  colorterm=  0
    nullchar= 128  linepattern= 16  linepage= 56
    baudrate=  4800  msecdial=  1000  msecrep=  1000  msecplt=  1500
    endpage-att
    7  45  45  62
c  BEL  -   -   >
c   terminal-init
    dialog-init
   27  58  27  12
c  ESC  :  ESC FF
    report-init
   27  58  27  12
c  ESC  :  ESC FF
    graphic-in-init
    7  27  26
c  BEL ESC SUB
    graphic-out-init
   27  59  27  12
c  ESC  ;  ESC FF
    terminal-end
   27  57  27  12
c  ESC  9  ESC FF
c
```

Each device table is made of up to 20 lines (plus comment lines beginning with character "c" in first column). Lines 4 to 20 are optional and depend of the device type considered.

Lines	Content	Max. number of characters
1	device name	32
2	device classes	keywords
3	switches	-
4	graphic parameters	-
5	graphic parameters	-
6	padding time parameters	-
7	endpage-att	keyword
8	endpage-att string	4
9	terminal-init	keyword
10	terminal-init string	32
11	dialog-init	keyword
12	dialog-init string	32
13	report-init	keyword
14	report-init string	16
15	graphic-in-init	keyword
16	graphic-in-init string	16
17	graphic-out-init	keyword
18	graphic-out-init string	32
19	terminal-end	keyword
20	terminal-end string	32

Line 1: This device name will appear in the select questions redifining the current devices.

Line 2: Device type (**terminal** and/or **printer** and/or **plotter)**

Line 3:

Switches number	Meaning
0	Graphic output
1	Graphic input
3	Hardware dashed vectors
4	Character size 1, 35L @ 74C
5	Character size 2, 38L @ 81C
6	Character size 3, 58L @ 121C
7	Character size 4, 64L @ 133C
13	Graphtext character generator
14	Color available
15	Visual 500/550
19	Write-thru mode
20	Cross hair positioning
21	Selective block erase

Line 4:

Number of extra bytes:	extrabytes = (0-3)
Arc smoothness:	arcsmooth = (0-3)
Soft character font:	charfont = (0-6)
	0 = ASCII
	1 = French
	2 = German
	3 = British
	4 = Swedish
	5 = Danish
	6 = Spanish
Type of terminal:	colorterm = (0-15)
	0 = Tektronix
	1 = HP2627A
	2 = Lexidata 2410
	3 = Seiko
	4 = Visual 550
	5 = Selanar
	6 = Plotdev

Line 5:

Null character:	nullchar = (ASCII value)
Number of line patterns or colors:	linepattern = (1-...)
Number of lines per page for reports:	linepage = (1-99)

Line 6: Synchronization parameters. If a normal baudrate is given, a certain number of padding (null) characters will be sent to the corresponding device when initializing it. This number correspond to

```
(baudrate * msecdial) / 1000  for dialog initialization
(baudrate * msecrep)  / 1000  for report initialization
(baudrate * msecplt)  / 1000  for plot initialization
```

Lines 7,9,11,13,15,17,19: Keywords of the different Tektronix-compatible command strings.

Lines 8,10,12,14,16,18,20: The command strings given as sequences of corresponding ASCII numeric value of the control characters.

2.2 NOTATION

This section contains the details of all the notation used in STAN to refer to different types of data. Each type of data is referred to by a well defined identifier such as a name, a number, etc. The user of STAN should know the notation that is employed for each data type in order to be able to use the system. For example, in a dialog which requires identification of a node of the base network, it is assumed that the user knows that a node is identified by its node number and that he will enter the appropriate number.

2.2.1 Network

A **scenario** is a set of data that defines a complete network in the STAN system. It contains the set of **modes, products, vehicles, base network** and **transfers.** Once an assignment has been performed on a scenario, all the results that are related to the network also become part of the scenario. The scenario identifier is a *number* between 1 and 9999. A title of up to 80 characters is stored with each scenario and is displayed with all the outputs produced with a given scenario, but it may not be used to access a scenario. At any time one scenario is defined to be the **current scenario.** Any reference made to network related data refers to the current scenario.

A transportation **mode** is identified by *one letter* that the user specifies when he enters the mode in the mode table. A description of up to 10 characters is stored with each mode and is displayed with certain outputs, but may not be used to access a mode.

A **product** is always referenced by a *product number* that the user specifies when he enters the product in the product table. The *product number* must be comprised between one and the maximum number of products, declared by the user when creating the STAN data base. A description of up to 10 characters is stored with each product and is displayed with certain outputs, but may not be used to access a product. **Vehicles** are specified by mode and by product. They are discribed by a transported weight, a vehicle weight and an optional convoy weight.

A **node** is always referenced by a *node number* which is defined by the user. Node numbers need not be contiguous but must be positive integers which cannot exceed 9999. All output that refers to nodes is sorted according to node numbers, with one exception: centroids always precede regular nodes.

A **link** is always identified by *an ordered pair of node numbers and a mode letter*, NO, M, ND, that indicate respectively the initial node, the mode used and terminal node of the link.

A **transfer** is denoted by a *quintuplet of node numbers and mode letters*, NA, NF, MF, NT, MT, where NA is the transfer node (which must not be a centroid), NF, MF, NA is the incoming link and NA, MT, NT is the outgoing link of the transfer.

2.2.2 Matrices

A **zone** is identified by the corresponding *centroid node number*. Sets of zones may also be specified by using zone groups.

A **zone group** is a set of zones that is denoted G*x*n where *x* is a letter of the alphabet which identifies the group ensemble Gx and *n* is a number between 0 and the maximum number of groups per ensemble. Each zone may belong to only one zone group of each ensemble Gx. Initially all the zones belong to group Gx0. The maximum number of zone groups is a compilation parameter.

A **matrix** is always denoted m*x*n, where *x* represents the type of matrix (f = full origin/destination matrix, o = origin vector, d = destination vector, s = scalar) and *n* is the matrix number, which may be an integer between 1 and the maximum number of matrices of each type that may be stored in a particular data base (this number may differ from one data base to another). With each matrix the user may also specify a six character name and a description of up to 40 characters. At present a matrix may not be accessed by its name, although it is probable that in a later release of STAN, the six character name of a matrix will also be used as an identifier (see also section 2.4.7).

A full **origin/destination matrix**, type **mf**, contains 1 element for each origin/destination pair. An **origin matrix**, type **mo**, contains one element for each origin. A **destination matrix**, type **md**, contains one element for each destination. A **scalar matrix**, type **ms**, contains **1 element** only.

At execution time, if data related to a matrix is read, but the data is not specific for origins or destinations, it will be automatically duplicated where necessary within the module. For example, if a full matrix is required in the module and the user specifies an origin vector, the data will be duplicated for all the destinations. If the data is written, and the module creates a matrix within the module, which is not specific for origins or destinations, it will be automatically duplicated where necessary.

2.2.3 Functions

The relation between base network and functions is given by the definition of function classes. A **function class** is an index that associates to a link (transfer) a triplet of functions: an operating cost, a time delay and an energy consumption function. Function classes are an efficient and compact way to associate functions to links and transfers. Several links (transfers) may share the same function class.

A **function** is a data structure which contains the algebraic expression of a functional relationship. A function is denoted F*tin*, where *ti* represents the function type and *n* is a number between 1 and the maximum number of functions per function type in the data base. Table 1 lists the function types that are used in STAN, their identifiers, and the keywords that denote variables which may be used in the function definition.

Functions are defined by means of algebraic expressions. Such algebraic expressions are also used for matrix calculations. An **algebraic expression** may contain:

- variable names (keywords and composite keywords)
- real or integer constants
- up to twenty levels of parentheses
- algebraic operators: +, −, *, / and ^ (exponentiation)
- relational operators:.EQ., .NE., .LE., .LT., .GE., .GT. (which yield 1 if "true", 0 if "false")
- minimum and maximum operators: .MIN., .MAX.
- modulo operator: .MOD.
- logical operators: .AND., .OR., .XOR. (for logical operators non zero values are considered "true" and zeros "false"; the results of a logical operation are always 1 for "true" and zero for "false")

- the following intrinsic functions:

ABS( )	absolute value
INT( )	integer (truncated)
EXP( )	exponential function
LN( )	natural logatithm
LGAM( )	loggamma function
SIN( )	sine, argument in radians
COS( )	cosine
ATAN( )	arc tangent
SQRT( )	square root

All operators are *left associative* and their precedence is defined as follows: exponentiation (^) precedes multiplication (*) division (/), logical product (.AND.) and exclusive or (.XOR.). The lowest precedence is given to addition (+), subtraction (−), logical or (.OR.), minimum (.MIN.) and maximum (.MAX.) operators, the modulo operator (.MOD.) and all relational operators (.LT., .LE., .EQ., .NE., .GE., .GT.). When the user wishes to perform operations in an order that does not correspond to these precedence rules, parentheses must be used.

A composite keyword is a special keyword where some predefined calculations (using simple keywords, see composed keyword TRAIN) are done when the evaluation of algebraic expressions is performed.

FUNCTIONS ON LINKS	
Function_Types:	**Codes:**
Operating cost	FCLnn
Delay	FDLnn
Energy consumption	FFLnn
Variables:	
LENGTH	Link length
CAP	Link capacity
BETA	Link special coefficient
UL1,UL2,UL3	Link user data 1 to 3
CONWGT	Convoy weight for current product on link
WBYVEH	Weight by vehicle for current product and for link mode
VEHWGT	Vehicle weight for current product and for link mode
UP1,UP2,UP3	Current product user data 1 to 3
VOLTOT	Total product volume on link
VEHPROD	Vehicles equivalent of current product volume on link
VEHTOT	Vehicles equivalent of total product volume on link
TRAIN	Convoys equivalent of total product volume on link
STRAIN	Convoys equivalent of total product volumes on parallel links that share the same infra structure with the mode of the current link
UM1,UM2,UM3	Link mode user data 1 to 3
UMP1,UMP2,UMP3	Link mode - current product user data 1 to 3

FUNCTIONS ON TRANSFERS	
Function_Types:	**Codes:**
Operating cost	FCTnn
Delay	FDTnn
Energy consumption	FFTnn
Variables:	
VOLTOT	Total product volume on transfer
UP1-UP3	Current product user data 1 to 3
UN1-UN3	Transfer node user data 1 to 3

GENERAL USE FUNCTIONS			
Function_Types:	**Codes:**	**Variables:**	
General for user	FGUnn	X0-X9	general variables

Table 1 Functions and Keywords

2.2.4 Modules

A **module** is a basic program unit of the STAN system, from the user's point of view. It is denoted by a number $x.yy$, where x is the module group and yy is the module number within the module group. A **module group** is a set of modules that has a particular role in the STAN system. See also section 3.

2.2.5 Demarcations

The user can specify up to nine sets of **demarcations** that may be superimposed on plots of the network. A set of demarcations is denoted by a *number between 1 and 9*. When several sets of demarcations are designed on a plot, the STAN system accepts a number composed of one digit for each set. For example 253 stands for the sets 2, 5 and 3.

2.3 MODULES

The modules of the STAN system are subdivided into six groups: Utilities, Network Editor, Matrix Editor, Function Editor, Assignment Procedures and Results.

Each module corresponds to a "stand alone" program that communicates (interfaces) with the other modules only through the data base. Table 2 is a list of all the STAN modules.

```
1.   U T I L I T I E S
1.11 Direct data base manipulations
1.21 Log book consultation
1.22 Scenario manipulations
1.23 Input / modify / output utility data
1.31 Input          demarcation lines using batch entry
1.32 Input / modify demarcation lines interactively
1.34 Output         demarcation lines
2.   N E T W O R K   E D I T O R
2.01 Input / modify / output modes
2.02 Input / modify / output products
2.11 Input          base network using batch entry
2.12 Input / modify base network interactively
2.13 Plot           base network
2.14 Output         base network
2.21 Input          transfers using batch entry
2.22 Input / modify transfers interactively
2.23 Plot           transfers
2.24 Output         transfers
2.33 Plot shortest paths on network
```

3. M A T R I X E D I T O R

3.01 Input / modify / display zone groups

3.11 Input matrices using batch entry

3.12 Input / modify matrices interactively

3.13 Plot matrices

3.14 Output matrices

3.15 Plot matrix histograms

3.25 Matrix calculations

3.26 Matrix balancing

4. F U N C T I O N E D I T O R

4.01 Input / modify / display function classes

4.11 Input functions using batch entry

4.12 Input / modify functions interactively

4.13 Plot functions

4.14 Output functions

5. A S S I G N M E N T P R O C E D U R E S

5.11 Prepare scenario for assignment

5.21 Multiproduct multimode assignment

5.31 Maxflow calculations

6. R E S U L T S

6.01 Result summary

6.13 Plot product costs and volumes

6.14 List product results

6.15 Plot product vehicles and convoys

6.23 Compare product costs and volumes

6.33 Plot shortest paths on network

9. E N D O F S E S S I O N

Table 2 The STAN Modules

2.3.1 The Menu

The user selects the next module that he wishes to execute, by means of the menu. The **menu** is a display of all the available module groups. Once a group is selected, the menu displays all the modules within that current group as well. The current module group always corresponds to the group that contains the last module that was called. The user is then asked for the next module that he wishes to call by means of the question:

Enter: Next module =

This question is somewhat special since it accepts as a valid answer either a module number or a module group number. If the answer is a group number, the menu is displayed again, this time giving all the modules that are available within that group, which becomes the new current group.

It is also allowed to enter a module number that does not belong to the current module group. The system will then call this module immediately.

If the user enters a module number which does not correspond to a standard module of the STAN system, the following error message is displayed:

Module x.yy is unknown!
You want to call it anyway?

If the user insists, by answering **yes** to the above question, the system assumes that the module exists (for instance, a special utility module that is specific for a given user, but is not part of the STAN system). A **no** answer causes the system to return to the menu and ask again for the next module number.

The menu also provides the possibility to change some important system parameters and to enter comments into the log book. The **menu commands,** which provide the means for carrying out these actions, are keywords that may be followed by numerical values or comments. The following menu commands may be used:

Command	**Action triggered by command**
device	The system generates a dialog which asks for the current device types for terminal, printer and plot file.
status	Display the status of the current scenario. The first three letters of the status indicate the status of the protection flags: **D**: protected against deletion **M**: protected against network modification **E**: protected against forced module execution The three following characters indicate the kind of assignment that has been performed on the scenario or for which the scenario has been prepared. One letter is provided for the multimode multiproduct assignment, the maxflow calculation and the last one, for futur developments, a - indicates that no assignment has been carried out; a **R** is displayed if the scenario has been prepared for the assignment, but has not been assigned yet; an **A** indicates that an assignement has been performed on the scenario.

page=<page no.> or short **p**=<page no.>	The current page number for report is reinitialized at the given value. If <page no.> is absent, the current page number is displayed.
scenario=<scenario> or short **s**=<scenario>	Change the current scenario to <scenario>. The number, description and status of the new current scenario are displayed. If no scenario number is specified, a list of all existing scenarios is shown.
comment=<log book comment> or short **c**=<log book comment>	Enter a comment along with the log book entry for the next module called. The comment may not exceed 60 characters.
on=<switch no.> **off**=<switch no.> **switch**=<switch no.> or short **sw**=<switch no.>	Set, reset or query the state of a given switch. Switches are numbered from 0 to 31 and are used to enable (**on**) or disable (**off**) certain features of the STAN system. If no switch number is entered along with the **switch** = command, the status of all switches is displayed. The following is a list of the switches used in the current implementation:

Bit	Action
0:	Ask for list device before each report.
1:	Ask for graphic device before each plot.
2:	Display menu.
3:	Ask for column-wise when a matrix is initialized.
4:	Ask whether module parameters should be saved on disk.
5:	Allow saving of predefined windows using the subcommand ">" of the graphic command **w**.
7:	Ask for terminal type at the beginning of a session.
8:	Ask for printer type at the beginning of a session.
9:	Ask for plot file type at the beginning of a session.
10:	Read device table at the beginning of a session (normally off).
13:	Allow shell commands using "~!".
14:	Copy mode, write simultaneously on plot file and terminal.
15:	Echo mode.
30:	Generate a message containing the compilation date when a module is entered (normally off).

Menu commands may be entered with a module number on the same line, in which case the module number must be given first, or without a module number, in which case the commands are executed and the question

Enter: Next module=

reappears. The only exception is the **comment=** command, which is only effective

when given on the same line, after the module number. More than one menu command may be entered on the same line, but they must be separated by commas or blanks. If a **comment=** command is followed by other commands, the log book comment must be given between quotes (") or must be exactly 60 characters long, in order to avoid the interpretation of the following command as part of the comment.

It is important to note that keywords and corresponding values must follow without intervening spaces (blanks) and that the module number may be preceded by an optional keyword **module=**, or short **m=**, thus permitting more readable directives when STAN is used in batch mode.

2.3.2 Calling a Module

Since STAN is conceived for main use in interactive mode, there is no predetermined sequence in which the modules must be executed. On the other hand, there are certain important considerations, that the user should be aware of, when choosing the next module:

a) as described in Chapter 1, the data items in the data base are structured in a certain hierarchy (see Figure 2 of Chapter 1). This implies that when a data element is added to the data base, all data elements it points to (higher in the hierarchy) must already be defined and that a data element may be deleted from the data base only if no data elements (lower in the hierarchy) are pointing to it.

b) the execution of some modules depends on the availability of certain data. Therefore, they may be called only when the current scenario is in a ready state and all the required data are available. The status of each scenario is given by a set of flags that are set and reset by the appropriate modules.

These flags are:

- scenario ready for multimode multiproduct assignment
- scenario ready for maxflow calculation
- product costs and volumes available

Whenever the user attempts to call a module for which the scenario is not in a ready state, an error message is displayed. An error message is also displayed when the user attempts to call a module which would reset a flag and result in destroying some of the information associated with the scenario (e.g. results of assignment), allowing thus the user to reconsider his decision. All STAN modules that modify data pertinent to an assignment reset the corresponding flags.

In order to allow the user to protect a scenario against an unwarranted modification or deletion, there are three flags associated with each scenario, which are controlled by the user:

- protect scenario against deletion
- protect scenario against network modification
- protect scenario against forced module execution

2.3.3 Module Parameters

Most of the modules contain certain parameters that are set to some default values which are not displayed and remain fixed unless the user wishes to change them. These parameters are referred to as the **module parameters** . This concept allows to diminish the size of the dialog to the essential module functions but it permits the user, whenever it is required, to control a variety of parameters that are not of importance in most instances.

The default values of the module parameters may be changed by the user. After the user has modified the module parameters of a given module the **Yes/No** question:

Save current values of module parameters on disk?

appears. If the user answers **yes**, the current values of the module parameters will become the default values for all subsequent calls of that module. Note that the default values of the module parameters are stored by **module**. This implies that a module parameter that applies to several modules may have different default values for different modules.

The above question, that allows redefinition of the default values of the module parameters, may be disabled once the user has set up "his" default values. This is done using switch no. 4 that is controlled by the menu commands **on=**, **off=**, **switch=**, as described in section 2.3.1.

All the modules that plot the network in one form or another contain a set of module parameters that give the user control over:

- The dimension of the boxes that represent nodes on the plot, both for displays that show the node number and for displays that do not show node numbers.
- The offset to the right when displaying links of the base network (a negative offset may be used to represent left hand traffic on a network).
- The size of the letters that are used to annotate links.
- The display of nodes that are not connected to the subnetwork that is plotted (default value: plot only nodes connected to the subnetwork).

The default values for the module parameters are chosen according to the graphic resolution of the Tektronix 4014 terminal and should yield plots of good readability

in most cases. However, if a user wishes to use the higher resolution made possible by some equipment, such as a plotter, in order to plot a larger section of the network on a single sheet, or if a user wishes to annotate a graphic output with big letters that are readable on 35 mm slides, the module parameters can be changed accordingly.

An example of a dialog for changing the module parameters associated with a network plot is given below:

Enter: Width,height of nodes without node number
(4.00, 4.00) = /
Enter: Width,height of nodes with node numbers
(32.00, 12.00) = /
Enter: Offset to the right for links (1.80) = /
Enter: Character height for text on links (6.00) = 10
Plot nodes only if connected to current link types?y
Save current values of module parameters on disk? y

2.4 SOME IMPORTANT DIALOGS

This section contains some important types of dialog that are used throughout the STAN system. All dialogs that are detailed in this section, will not be described again in the presentations of each module which uses them. Each of the dialogs described in the following are generated by a subroutine of the STAN system, rather than by a module, and all the error messages that correspond to these parts of the dialog have error numbers that are larger than 1000. These error messages are documented in the first section of Appendix A of the STAN Manual.

The subsections that follow have as title the main question of the type of dialog they are part of, which is usually also the first question that the user is confronted with.

2.4.1 Display node numbers ?

On all network plots, the nodes are represented by rectangular boxes. Centroids are distinguished from regular nodes by the fact that their boxes are drawn twice, yielding the visual impression of a "fatter" box. Node boxes may be empty or may contain the node number. In both cases, the node box is kept clear from incident link, i.e. only the part of the link that does not overlap with the boxes of its *extremal nodes* are plotted. This technique only works properly if the node boxes do not overlap.

The question

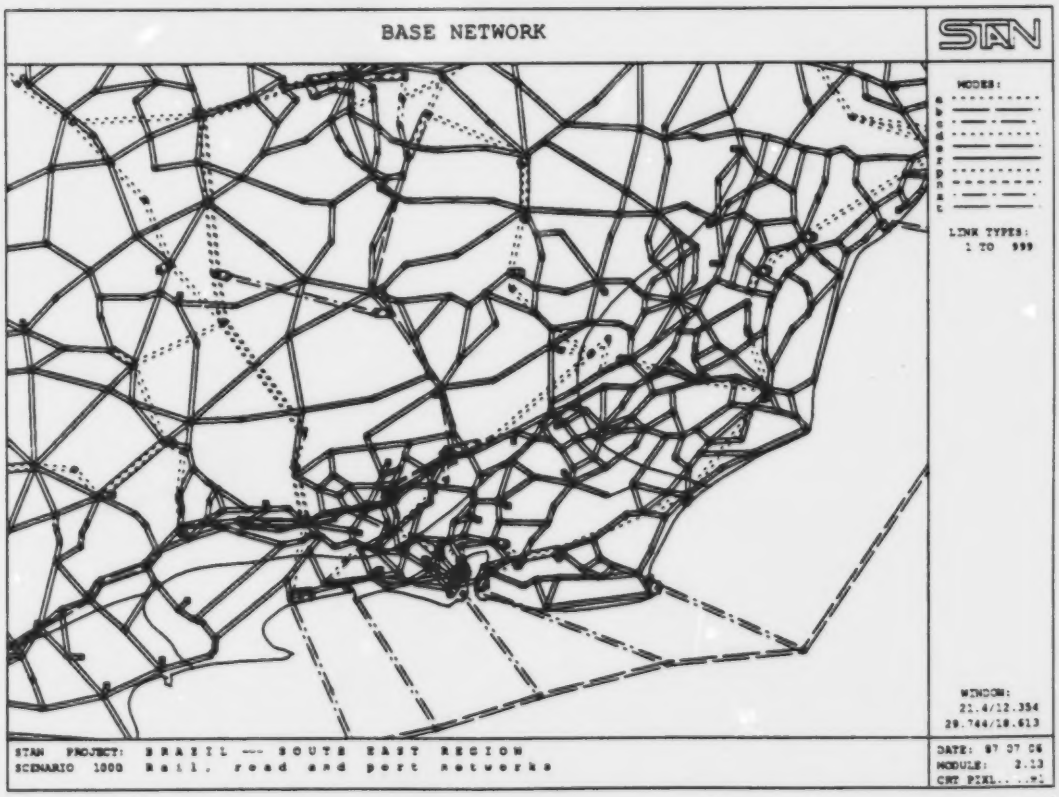
Display node numbers?

allows the user to choose to display or not the node numbers. The size of the node

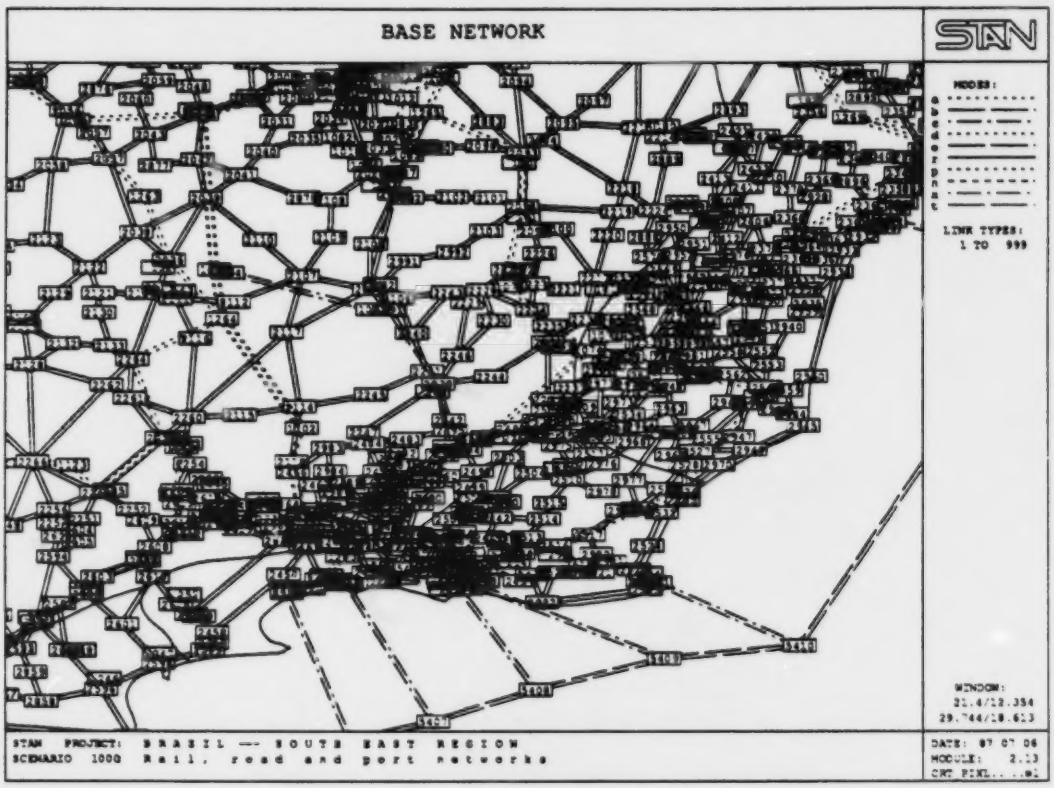
boxes on the plot is considerably larger when node numbers are displayed (unless the module parameters have been set otherwise - see section 2.3.3). Thus, the user is well advised to display node numbers only when a relatively small part of the network is plotted, otherwise the plot becomes unreadable, because of overlapping node boxes (cf: Example b).

The above question determines only the format of the node boxes for the plot that immediately follows the dialog. Subsequent plots that are produced by graphic commands do not necessarily display the nodes in the same way. The graphic command N, which may be used as a "toggle switch", permits the user to choose to display or not the node numbers for the next plot (see section 2.5 for details).

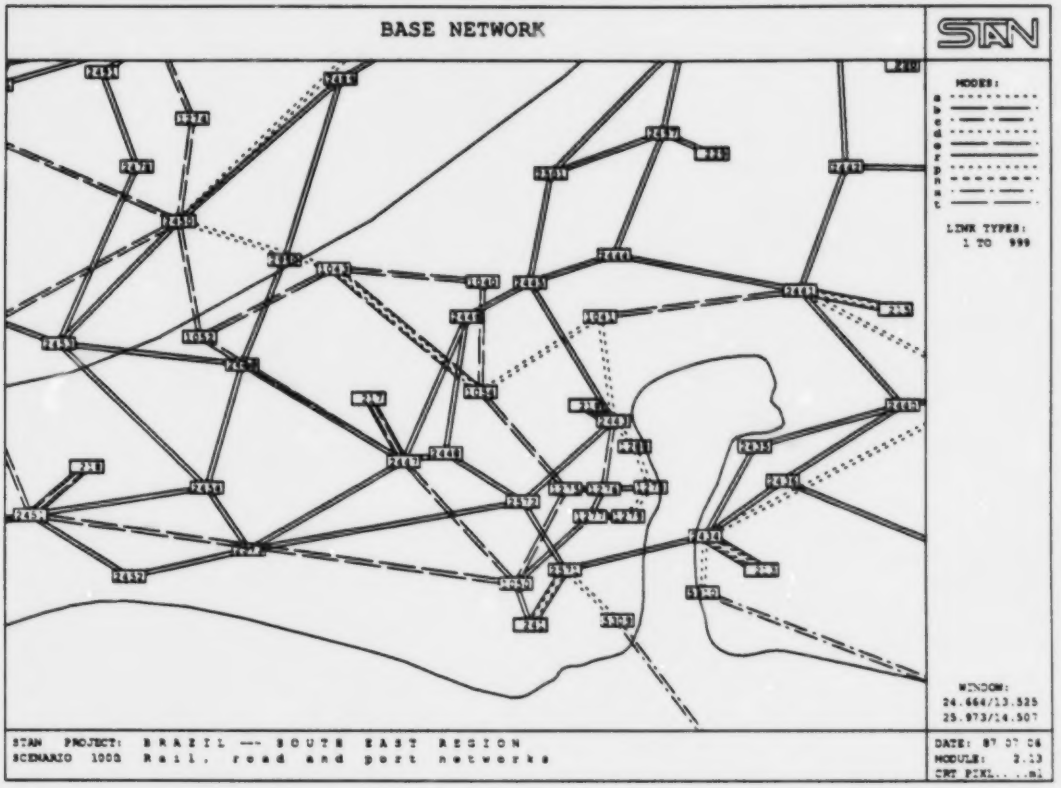
Example a): No node numbers are displayed and a relatively big part of the network is plotted:



Example b): The same part of the network as in example a) is plotted, but this time the node numbers are displayed. The resulting plot is unreadable, since the nodes are too close together and their boxes overlap.



Example c): A small part of the network is displayed in detail, together with the node numbers. The level of the detail chosen results in a clearly readable output.



2.4.2 Enter: Mode(s)=

All modules of the STAN system which require the specification of a subset of modes employ the question

Enter: Mode(s) =

The user is expected to enter either a list of the desired modes, by giving, in any order, the one letter abbreviation for each mode, or * to indicate all modes. The mode abbreviations may be separated by commas or blanks or may be concatenated as a single string of characters. Example: "abc", "c,b,a" or "b a c" are all equivalent answers.

If the user's answer contains an undefined mode, the following message is displayed:

Allowed modes are: a b c d e r p n x t

Enter one or more modes on the same line or * for all modes!

and the user is expected to enter a valid subset of modes. Note that a blank line is also a valid answer, and is interpreted as the empty set of modes, that is, no modes are specified.

2.4.3 Enter: Nodes (from,to)

This multiple **Enter** question is used whenever a subset of nodes is to be defined by the user:

Enter: Nodes (from,to)

=< n_1 >

=< n_1 >,< n_2 >

=< n_1 >,< n_2 >,<**incr**>

=/

The following alternative input types are allowed on each line:

a) one node number: this node is to be included;

b) two node numbers: all the nodes in this interval are to be included;

c) two node numbers and an increment: all nodes with node numbers of the form n_1, n_1+incr, n_1+2*incr, ... in the given interval are to be included;

c) * for all nodes.

2.4.4 Enter: Desired link types (from, to)

All modules of the STAN system, which output the network in one form or another, provide the possibility to output only a selected subnetwork. A subnetwork is defined as all the links that have link types specified by the user while answering the question:

Enter: Desired link types (from,to)

=< l_1 >

=< l_1 >,< l_2 >

=< l_1 >,< l_2 >,<**incr**>

=/

This multiple line **Enter** question (see also section 2.1.2) allows the user to enter on each line one of the following alternatives:

a) one number, which represents a single link type that is to be included in the selected set of link types;

b) two numbers, which represent the lower and the upper limit of a link type interval that is to be included in the set of selected link types (including the limits);

c) three numbers, l_1, l_2, l_3, which causes all link types of the form l_1, $l_1 + l_3$, $l_1 + 2l_3$, $l_1 + 3l_3$, ..., that do not exceed l_2, to be included in the selected set of link types;

d) the same as a), b) or c), but the first number is preceded by a minus sign, which causes the specified link types to be excluded from the current set of selected link types;

e) * for all links types.

Link types must be defined by using numbers between 1 and 999. In the exmple that follows the user selects the link types from 20 to 69, but excluding link types 30, 40, 50 and 60, and including link type 12 as well:

Enter: Desired link types (from,to)
= 20,69
= -30,60,10
= 12
= /

2.4.5 Enter: Product numbers (from, to)

All modules of the STAN system, which need products to perform some calculation or graphic output, provide the possibility to select a subset of products by means of the question:

Enter: Product numbers (from,to)
=< p_1 >
=< p_1 >,< p_2 >
=< p_1 >,< p_2 >,<incr>
=/

This multiple line **Enter** question (see also section 2.1.2) allows the user to enter on each line one of the following alternatives:

a) one number, which represents a single product number that is to be included in the selected set of products;

b) two numbers, which represent the lower and the upper limit of a product number interval that is to be included in the set of selected products (including the limits);

c) three numbers, p_1, p_2, p_3, which causes all product numbers of the form p_1, $p_1 + p_3$, $p_1 + 2p_3$, $p_1 + 3p_3$,...., that do not exceed p_2, to be included in the selected set of products;

d) the same as a), b) or c), but the first number is preceded by a minus sign, which causes the specified product numbers to be excluded from the current set of selected products;

e) * for all products.

Product numbers must be defined by using numbers between 1 and the maximum number of products (a compilation parameter).

2.4.6 Window?

When a network plot is asked for in the STAN system, the user may always define a window to select only a part of the network. This is done by using the **Yes/No** question:

Window?

If the user answers **no**, the window is adapted automatically to contain entirely the selected subnetwork. If the user answers **yes**, the following **Select** question is generated:

Select: 1 = using coordinates
2 = using nodes
3 = centered on a node
4 = use predefined window
5 = use same window as before

If the user selects to enter window coordinates (1), he is expected to provide the coordinates of the lower left and the upper right corners of the desired window, by replying to the questions

Enter: Lower left corner (x,y) = *xll, yll*
Enter: Upper right corner (x,y) = *xur, yur*

If the X:Y ratio of the selected window does not correspond to the X:Y ratio of the actual plotting surface (4:3), the window is automatically adjusted by the system.

If the user wishes to define a window by using nodes (2), he is expected to specify a set of nodes by means of the multiple **Enter** question:

Enter: Nodes defining window (max. 10 per line)
=

The set of nodes that is entered in this way is then used to compute the window coordinates that include all the specified nodes at the largest scale possible.

The user may also define a window by specifying a center node and the diameter of the window (3). This is done by answering the sequence of two questions

Enter: center node (O =center of default window) =
Enter: diameter of window (xxxxxx) =

The center node will be located in the center of the plot. If a 0 is entered instead of a node number, the center of the plot is taken to be the center of the default window, which is the window that contains the entire subnetwork. This allows the user to obtain plots at a given scale (diameter). In this case the diameter of the window must be at least the value given in parentheses, which corresponds to the diameter that includes exactly the entire subnetwork, in coordinate units since otherwise only part of the subnetwork will be displayed.

The window may also be established using a predefined window (4), i.e. a window that has previously been saved using the subcommand '>' of the graphic command **W** (see section 2.5.3). Since the predefined windows are denoted 'a' to 'z', the user has to enter the the corresponding letter as an answer to the question

Enter: Window (a-z)=

An error message is displayed if the desired window is not defined.

The user may also select to display the same window that was used for the most recent network plot (5). The most recent plot is not limited to the plots generated in the current call to the current module, but may refer to a plot in another module or even during a previous session. In this case, no further dialog is generated.

In the case where no default window can be established (e.g. no links are yet defined, or an empty subnetwork has been defined), the system generates the message:

No default window can be established,
window must be defined manually

which is followed by the **Select** question explained above.

2.4.7 Enter: Origins (from, to)

Whenever a module in the STAN system requires the specification of a set of zones, one of the following questions is used, depending on the context.

Enter: Origins (from,to)
=

Enter: Destinations (from,to)
=

Enter: Zones (from,to)
=

This multiple line **Enter** question (see also section 2.1.2) allows the user to enter

one of the following alternatives on each line:

a) one zone number, which specifies that the zone is to be included in the set of zones

b) two zone numbers, which specify the lower and upper limits of a zone number interval; any zones within the interval, including the limits, are included in the set of zones

c) one group, which specifies that all the zones in the group are to be included in the set of zones

d) two groups from the same ensemble, which specify the lower and upper limits of a group interval; all the zones in any group within the interval, including the limits, are to be included in the set of zones

e) **all**, which specifies that all zones are to be included in this set of zones; **all** terminates the specification of the set of zones

f) **same**, which specifies that this set of zones will be the same as the set of zones specified the last time the identical question was asked in this module; if no such set has been specified, the set is empty and the answer **same** will result in an error message **no zones in list** followed by the **Enter** question. For example, if the current question is

Enter: Origins (from,to)

and **same** is entered, the set of zones will be that specified in answer to the preceding **Enter: Origins** question in this module, or an empty set if there was no such preceding question; **same** can only be entered on the first answer line and it terminates the specification of the set of zones unless the set is empty

g) **origins**, (or **o**), which specifies that this set of zones is to include the set of zones which was specified in the last **Enter: Origins** question in this module; if there was no such preceding question, **same** in that additional zones can be specified on preceding and following answer lines

h) **destinations**, (or **d**), which specifies that this set of zones is to include the set of zones which was specified the last **Enter: Destinations** question in this module; otherwise it is similar to **origins**

i) **zones**, (or **z**), which specifies that this set of zones is to include the set of zones which was specified in the last **Enter: Zones** question in this module; otherwise it is similar to **origins**

These questions are often preceded by the question:

Submatrix?

Answering **no** to this question means that the whole matrix will be used and the above questions are not asked.

In the following example, the user wishes to specify a set of origins which include those in group **ga0** plus origins 35 through 40 and origin 69.

Enter: origins (from,to)
=ga0
=35 40
=69
=/

2.4.8 Enter: Matrix

All the modules of the STAN system which require specification of a matrix use the question:

Enter: Matrix =

If a specific type of matrix is required (see section 3.2) this is indicated in the question as follows:

Enter: Matrix (mt) =

where *t* is the matrix type.

If one or several matrix types can be used. this is indicated as a list of types in the question, for example:

Enter: Matrix (md mf) =

The user is expected to enter a matrix identifier but he may also request a display of all or part of the matrix directory before entering a matrix identifier. If **m** is entered, a list of all the current matrices is displayed. If **ms**, **mo**, **md** or **mf** are entered the current matrices of that type are displayed.

Note that if the user enters a blank or a slash (/) instead of a matrix identifier, this is equivalent to specifying no matrix. Subsequent action depends on the module.

If information is to be read from the matrix specified, this dialog ends. Otherwise the dialog continues depending on whether the matrix already exists or not:

- the matrix does not exist: the user is notified and the question

Do you want to initialize it?

appears. If the user answers **no**, the **Enter: Matrix=** question will reappear. If the user answers **yes**, the questions

Enter: Name (6 chars) =
Enter: Description (40 chars) =

appear. The user is expected to enter the matrix name and description, which form the matrix directory entry. If switch 3 is **on** (see menu commands, section 2.3.1) the question

Column-wise?

will appear. If the answer is **yes**, the matrix will be stored column-wise, i.e. by destination. Then the question

Enter: Default value =

is displayed and the user must enter the value used to initialize all the matrix elements. With a blank answer, the default value is zero.

- the matrix exists: the question

Change header information?

is displayed. If the user answers **no**, the matrix will be used as it is and this dialog ends. If the answer is **yes**, the questions

Enter: Name (6 chars) =
Enter: Description (40 chars) =

appear and the user may redefine the matrix name or its description, or enter blank lines for the items he does not wish to change. The user may initialize the matrix elements of the specified matrix by answering **yes** to the question

Initialize data?

In this case the user is prompted to enter the value to which the elements are initialized by the question

Enter: Default value =

A blank answer to this question results in a zero default value.

2.4.9 Enter: Function

Two different forms of this question are used. If only one function is to be identified, the single line **Enter** question

Enter: Function = < *function*>

is asked. It accepts as an answer any function identifier of the form F*tin*, where *ti* stands for the function type and *n* is the function number.

If a set of functions is to be selected, the multiple line **Enter** question

Enter: Functions (from,to)
=< *function* >
=< *function* >, < *function* >
=/

The answer on each line may be either a single function identifier, or an interval of functions of the same class.

The user can also specify all defined functions of all function types by means of the following answer:

Enter: Functions (from,to)
=*

2.4.10 Constraint matrix

Most of the modules that perform matrix operations allow the use of a *constraint matrix* and the associated *constraint interval*. Most of the operations which involve matrices are normally performed on the entire matrix or on a submatrix defined by a subset of origins and or a subset of destinations. In addition, the matrix elements to be considered may be selected by means of a constraint matrix and a constraint interval. In this case only those elements, which correspond to elements of the constraint matrix that satisfy the specified constraint interval, will be considered.

The interval is specified by its lower limit (x_1), upper limit (x_2) and condition (Include or Exclude).

If **Include** (**Exclude**) is specified, an element is included (excluded) if the corresponding constraint matrix element (x) is within the interval, that is:

$$x_1 <= x < x_2$$

This allows the user, for example, to limit a matrix operation to only those elements that have positive demand. In this case the constraint matrix would be a suitable demand matrix and the interval [0.001,9999.0,Include] for the valid elements. The interval always includes its lower limit and excludes its upper limit.

All the modules of the STAN system which allow a constraint matrix use the question

Constraint matrix

Enter: Matrix =

to allow the user to specify a constraint matrix. The **Enter: Matrix** question is described in section 2.4.7. If the user enters a blank or slash (/), this indicates that no constraint matrix is required. If a matrix is specified, the user is asked to specify an interval by answering the **Enter** question:

Enter: Constraint interval (x_1, x_2, *condition*) =

where x_1 = lower limit of the current constraint interval
x_2 = upper limit of the current constraint interval
condition = "Include" or "Exclude"

The user is expected to specify the lower limit (x_1) and the upper limit (x_2) of the required interval together with the condition on the interval. Include may be abbreviated to **in** and Exclude to **ex**.

After a module has been executed and a constraint matrix specified, if a constraint matrix is required again, the user is asked:

Same constraint matrix *(matrix)* **and interval?**

where *matrix* is the current constraint matrix. If the answer is **no** the questions described above will be used again to allow the user to define a new constraint matrix and interval.

2.5 GRAPHIC OUTPUT

2.5.1 Graphic Display Formats

There are two main types of graphic displays that are generated by the STAN system: the **graphic work sheet** and the **plot**.

The graphic work sheet is used whenever some of the contents of the data base is to be modified in an interactive graphic way. A typical example, from the network editor, is given in Figure 2. A graphic work sheet is subdivided into four areas, which are also indicated in Figure 2:

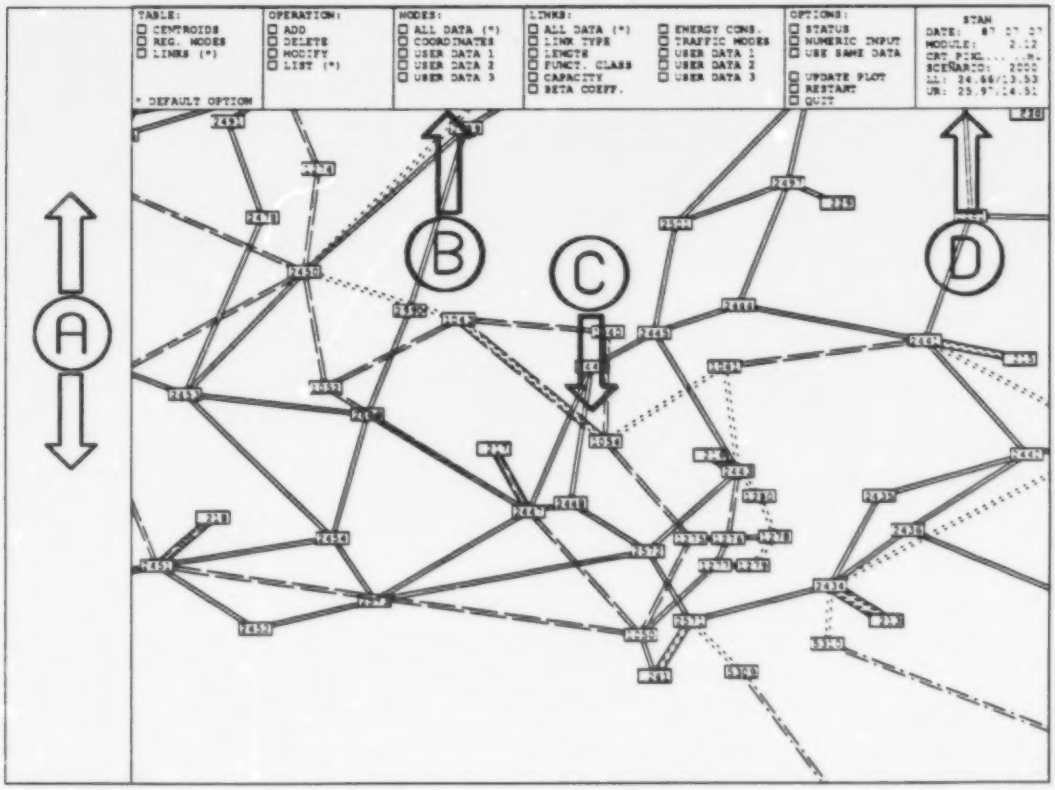


Figure 2 Graphic Work Sheet

A) The left margin is an **alphanumeric pad** of 16 characters width, which is used to display short messages and to input alphanumeric items by using an abbreviated dialog. Thus, this pad contains a record of the actions taken by the user. When the left margin is filled all the way to the bottom, the pad is continued on the plot area (C) from left to right. The user may avoid such an overflow by requesting an ***update plot*** (B) which will (among other) clear the left margin.

B) The **crosshair cursor command** area is located at the top of the work sheet. It contains an annotated box for each command. In order to execute a command, the graphic input device, which is referred to in the following as the crosshair cursor, is positioned into the corresponding box and a key is depressed. The meaning of the commands and the keys that are to be depressed (usually just a space) are specific to each module that uses a graphic worksheet.

C) The main part of the display is used as the **actual plot area** of the graphic work sheet, for displaying, entering and modifying the relevant data.

D) The upper right corner of the work sheet always contains the **date**, the **module number**, the **license number**, the **user initials**, the **current scenario** and the **window coordinates**.

The second type of graphic display, the **plot**, is used whenever a graphic output is produced or when a data base is queried in an interactive graphic way. Figure 3 is a typical example of a plot; the standard parts of the display are annotated A) to F).

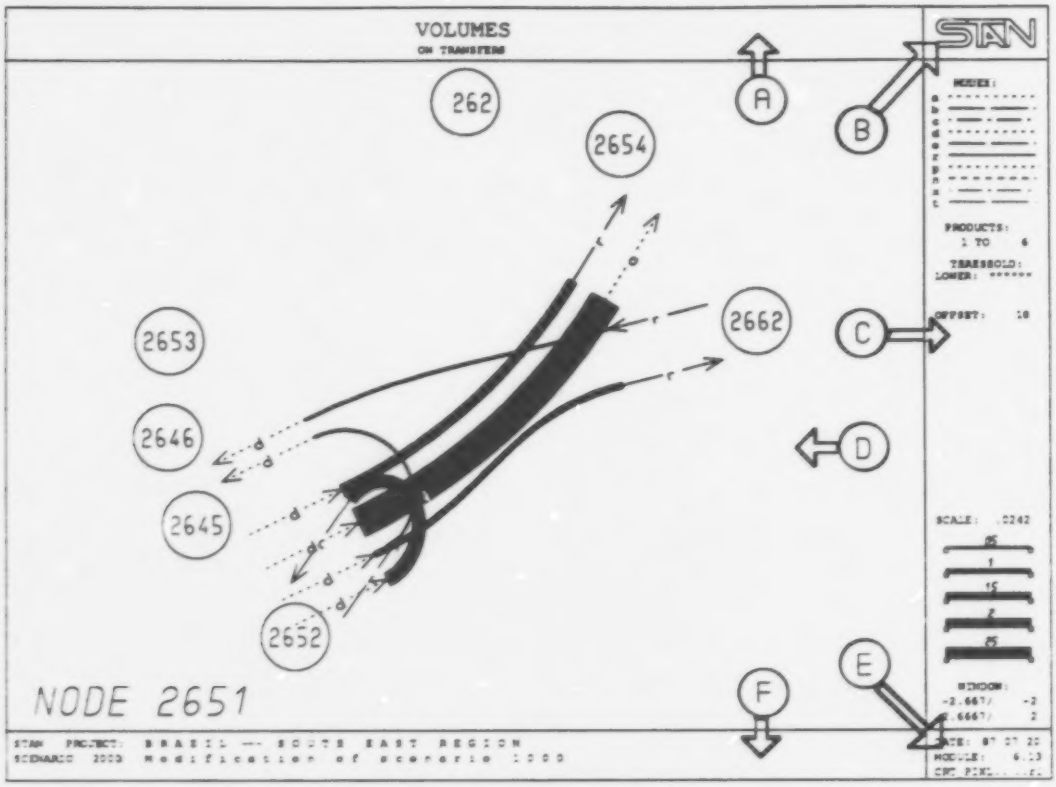


Figure 3 Typical Plot

A) The top of the plot contains a title, which is a description of the type of data that is displayed.

B) The STAN banner appears in the upper right hand corner of all plots.

C) The right margin is used to display the pertinent parameters that were used to generate the plot. In the case of network plots, it always contains the modes and the link types that were used to define the subnetwork and the coordinates of the window.

D) The main part of the display is the actual plot area. In the case of network plots it contains the window, that is the subnetwork selected for display.

E) The lower right corner contains date, module number, installation name and the user's initials.

F) The bottom of the plot contains a description of project title and the pertinent scenario and/or matrix description.

2.5.2 Graphic commands

When plots are generated interactively on the screen of a graphic terminal, the bell sounds when the plot is completed and the cursor moves to the graphic command position, to the right of the title box (F) at the bottom of the plot. The STAN system is then ready to accept graphic commands. A graphic command is denoted by a single letter or number. One or more (max.10) graphic commands may be entered on the same line. Tables 4a and 4b show a list of the available graphic commands.

Command:	*Action:*	
C	Continue on the same plot. Only valid when explicitly stated.	
M	Enter the MANUAL mode for annotating a plot manually.	
	Subcommands:	
	1,2,3,4	Text of different sizes
	A	Start polygon using full lines
	B	Start polygon dotted lines
	C	Start polygon dot-dashed lines
	D	Start polygon short dashed lines
	E	Start polygon long dashed lines
	(blank)	Next point on polygon
	G	Exit manual mode, back to level of graphic commands
N	*Toggle* switch for displaying or not displaying node numbers.	*
O	Redefine the transfer offset for the display of transfers at transfer node. The question OFFSET= appears at the bottom of the plot.	**
P	Replot the same plot on the background graphic device (plot file). This option is not available on all installations.	
Q	Quit	leave this module and return to the menu.
R	Restart	return to the beginning of the module.

(* : only valid for network plots)

(** : only valid for transfer plots)

Table 4a Graphic Commands

Command:	*Action:*	
S	Redefine the scale for the display of attributes. The question SCALE= appears next at the bottom of the plot. Available for all plots that show a scale legend in the right hand margin.	
T	Pick some transfer nodes on network to be plotted or pick centroids in shortest path interactive queries, subcommands are:	*
	(blank)	To select graphically a transfer node or a centroid
	1,...9	To indicate how many of transfer nodes will be displayed on following plots, at this time typing 1,...9 will finish the transfer node selection
	G	To return to graphic command level
	N	To select by numeric input a transfer node or a centroid
	Q	To quit the module
	R	To restart the module.
W	Enter the WINDOW command. Subcommands are:	*
	(blank)	Lower left, upper right corners (with crosshair cursor)
	C	Define center of window
	U,D,L,R	Up, down, left, right
	1,...9	Zoom out n times
	>x	Save current window as predefined window x (x: letters a-z)
	<x	Recall predefined window x
	^	Use previous window
1 - 9	Plot demarcation set n (n=1,2,..9)	*
Blank	Continue (usually same as R)	

(*: only valid for network plots)

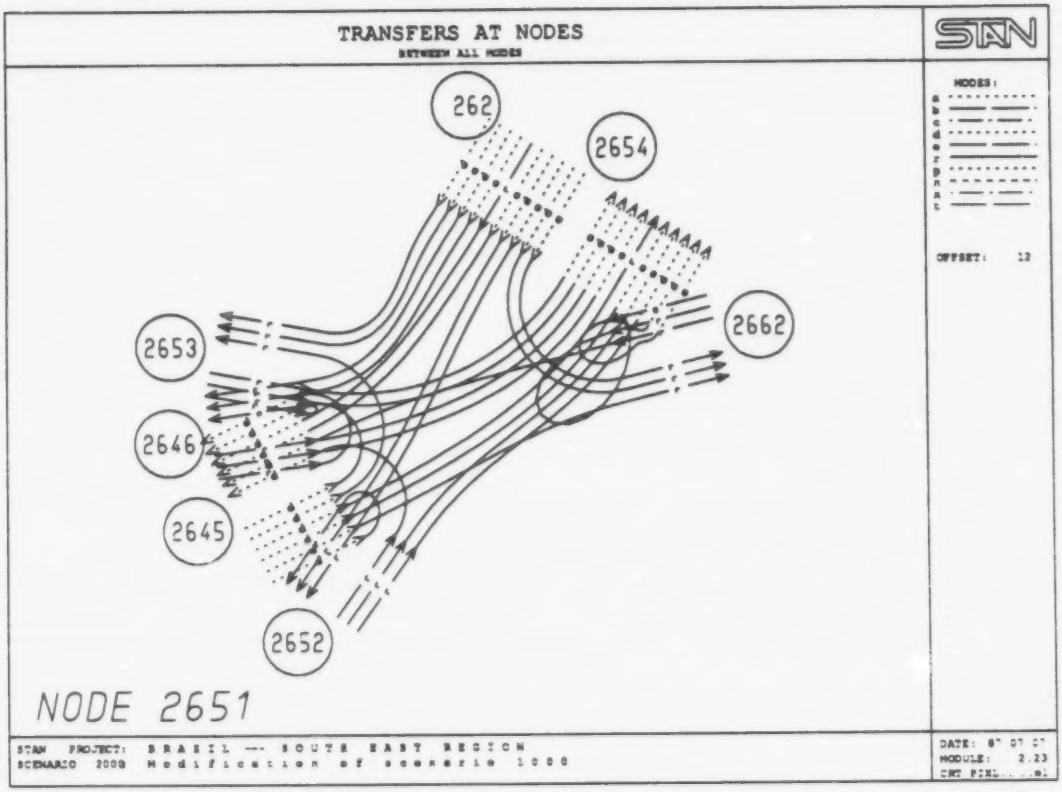
Table 4b Graphic Commands

The **C command** is used when the user wishes to continue to work on the existing plot. This command is only allowed in certain cases in which the plot is established in several steps (for examples see modules 2.33 and 6.33).

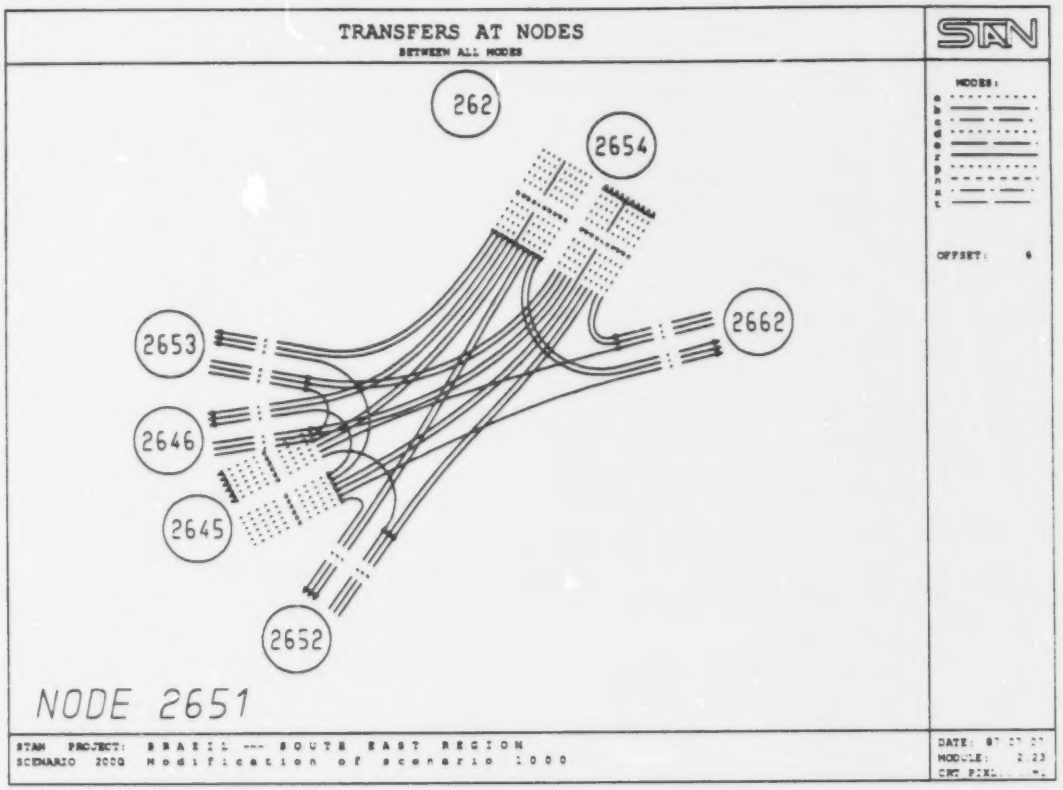
The **M command** allows the user to annotate a plot by manual intervention, by permitting the addition of text and drawing. The **M** command is described in detail in section 2.5.4.

The **N command** is effective only for network plots. It allows the user to remove or add the node numbers on a plot. If no node numbers are displayed on the current plot, the new plot will display the node numbers and vice versa. The **N** command is often used in combination with the W command, when changing the size of the window. This command is only allowed in network plots.

The **O command** is used with plots which display transfers at a node. The offset value (in hardware units) currently used is entered by the user to the question **OFFSET=** that appears at the bottom of the plot. The offset represents the distance between two adjacent transfers. The following example illustrates the use of the **O** command: after the first plot, an O command is entered and the user defines OFFSET= 6.


TRANSFERS AT NODES
BETWEEN ALL MODES
STAN
262
2654
2653
2662
2646
2645
2652
NODE 2651
NODES:
OFFSET: 12
STAN PROJECT: BRASIL --- SOUTH EAST REGION
SCENARIO 2000 Modification of scenario 1000
DATE:
MODULE: 2.23

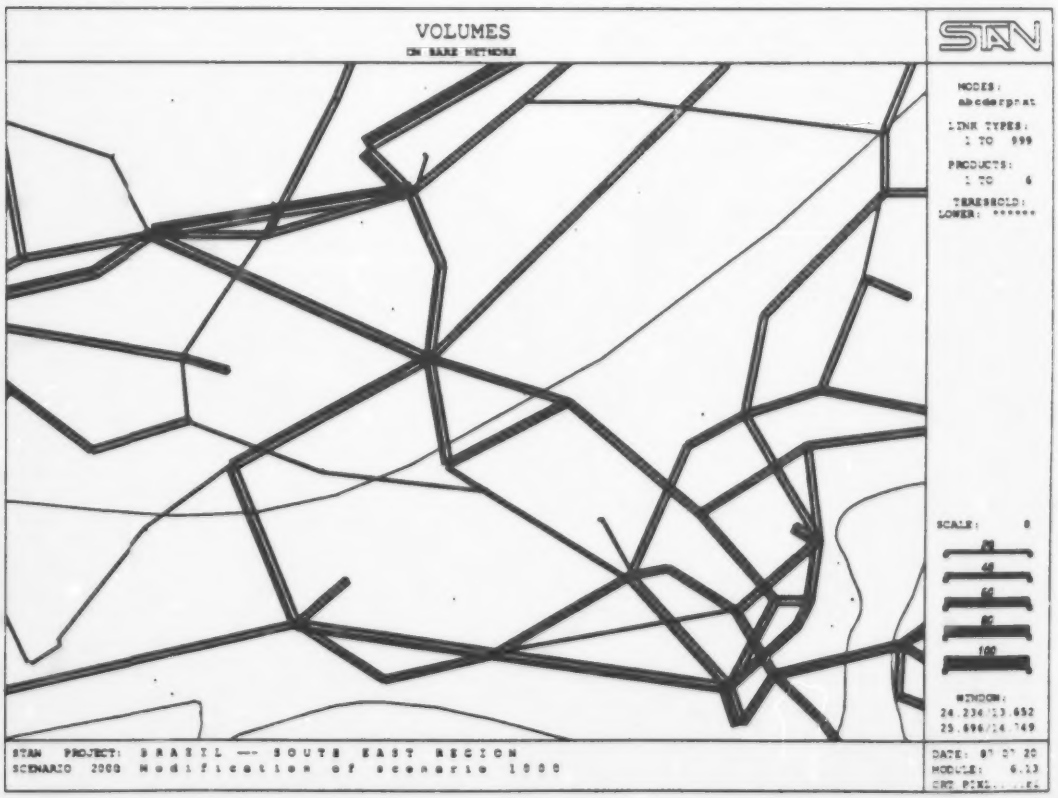
The plot is immediately redrawn, but this time the distance between transfers is smaller to permit to see more accurately all transfers at this node.



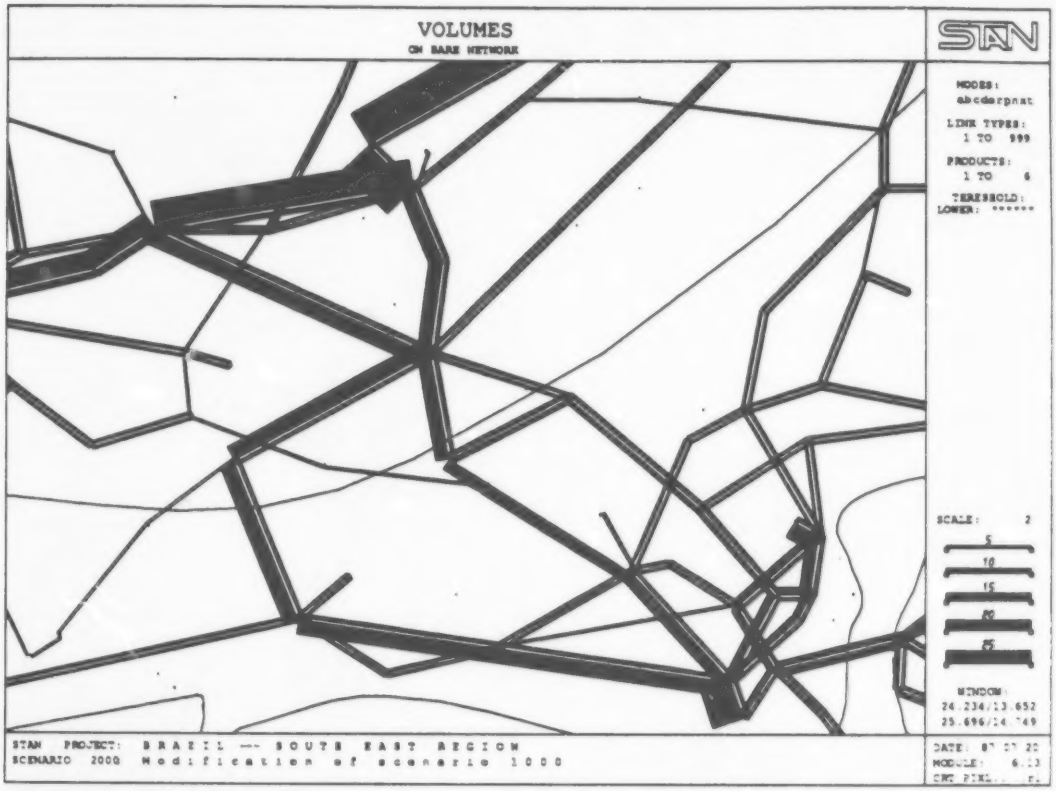
The **P command** generates the current plot on the background graphic device (plot file). This command is not supported on all installations and is never valid for graphic worksheets.

The **Q and R commands** generate the same response as elsewhere in the STAN system: return to the menu (with **Q**) or to the beginning of the current module (with **R**).

The **S command** is used with plots which display data in the form of bars or columns. The scale currently used is changed by a factor that is entered by the user to the question **SCALE=** that appears at the bottom of the plot. The size of the bars, circles or columns, is obtained by the number which was entered; this value represents the data value associated with each hardware unit on screen. The following example illustrates the use of the **S** command: after the first plot, an S command is entered and the user defines SCALE= 2.



The plot is immediately redrawn, but this time the bars, that are proportional to the link volumes, are four times as wide.



The **T command** is used to pick nodes within two different contexts: to display transfers at nodes or to plot shortest paths between chosen centroids.

In modules 2.23, 6.13 and 6.23 a selection of one or several transfer nodes is asked by the **T** command. The selection is then done by positioning the cursor on each of the chosen nodes and by depressing the space bar.

In crosshair cursor mode, the user has the choice to type

space bar : to select graphically a transfer node.

1 - 9 : to indicate how many transfer nodes will
be displayed on following plots;
this ends the transfer node selection.

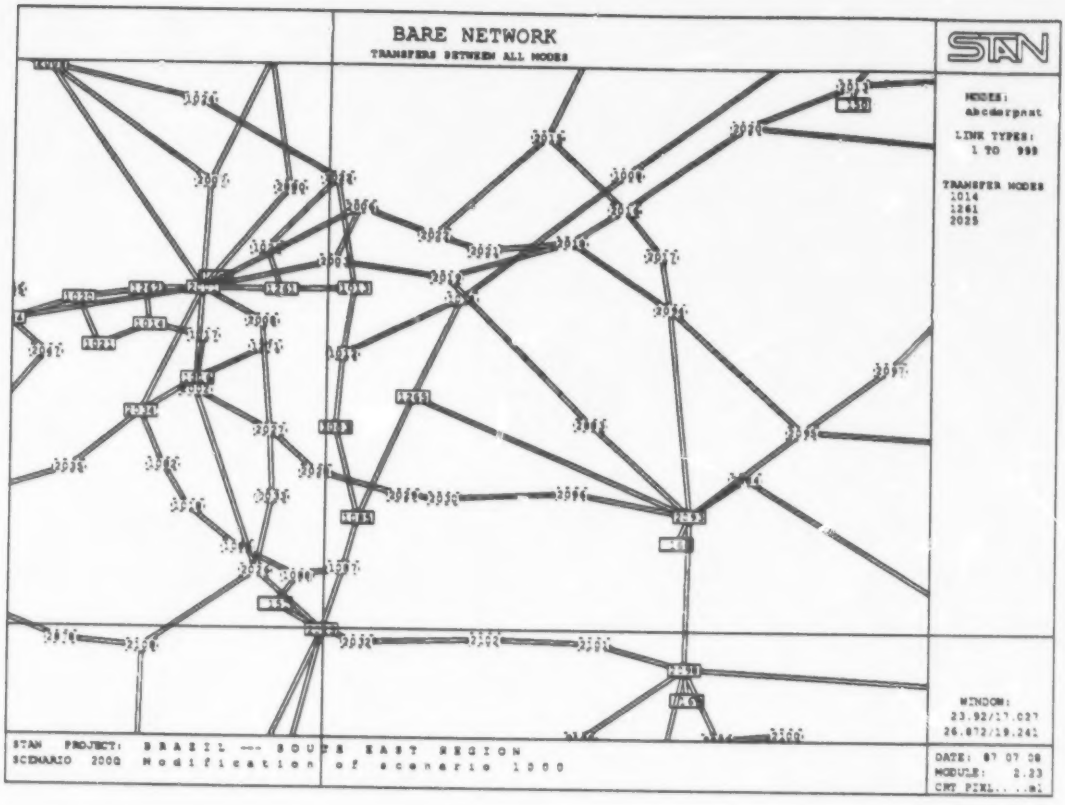
G : to return to graphic command level.

N : to select by numeric input a transfer node.

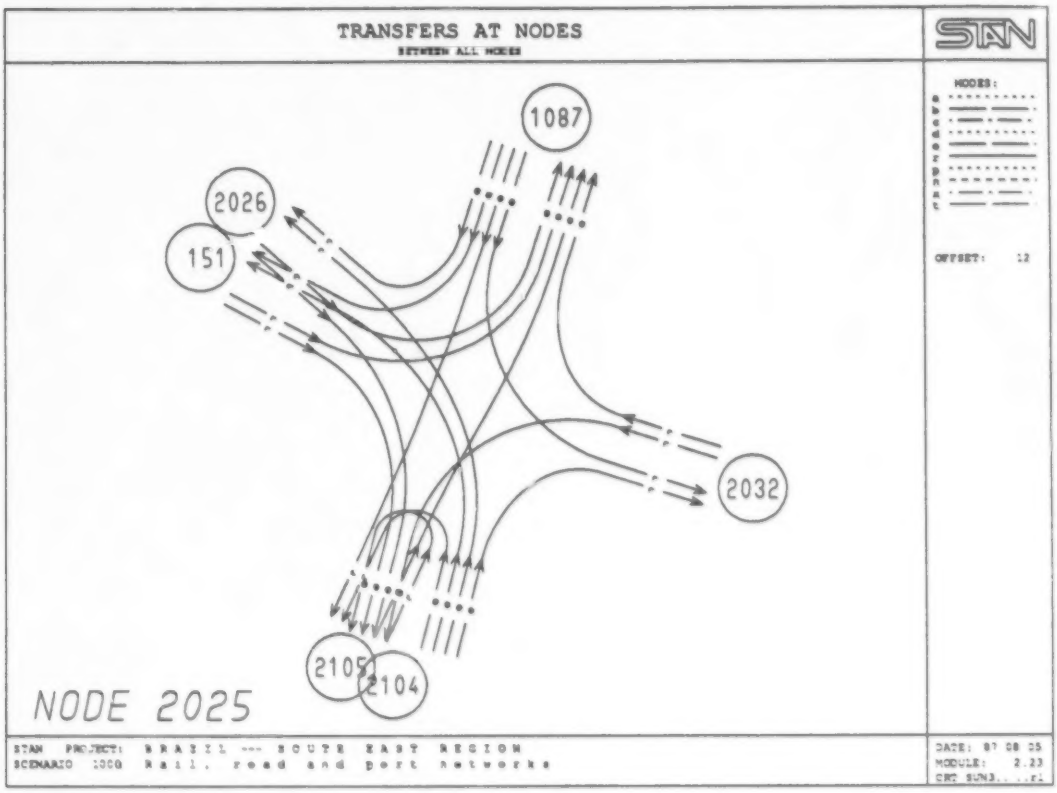
Q : to quit the module.

R : to restart the module.

The following example illustrates the use of the **T** command: after the selection of three transfer nodes, the user types 1, to see one transfer node per plot page.


BARE NETWORK
TRANSFERS BETWEEN ALL MODES
STAN
MODES:
abcdefgnst
LINK TYPES:
1 TO 999
TRANSFER NODES
1014
1261
2025
WINDOW:
23.92/17.037
26.872/19.241
STAN PROJECT: BRAZIL --- SOUTH EAST REGION
SCENARIO 2000 Modification of scenario 1000
DATE: 87 07 08
MODULE: 2.23
CMT FIXL...mi

The following plot represents one of the three selected transfer nodes from the preceding example:



In modules 2.33 and 6.33, the **T** command is used to select a centroid as a destination of shortest paths from a given root. After typing T, the user selects centroids by positioning the crosshair cursor on the centroid and by depressing the space bar.

In cross-hair cursor mode, the user has the choice of types

space bar : to select graphically a centroid.

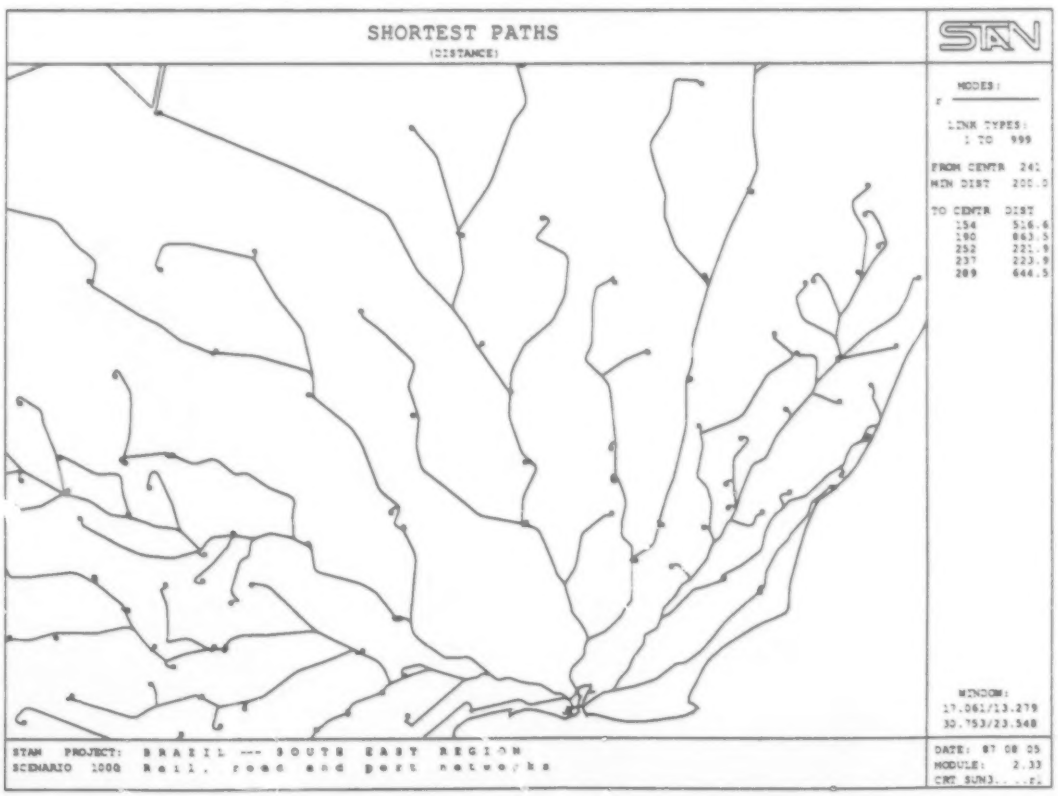
G : to return to graphic command level.

N : to select by numeric input a centroid.

Q : to quit the module.

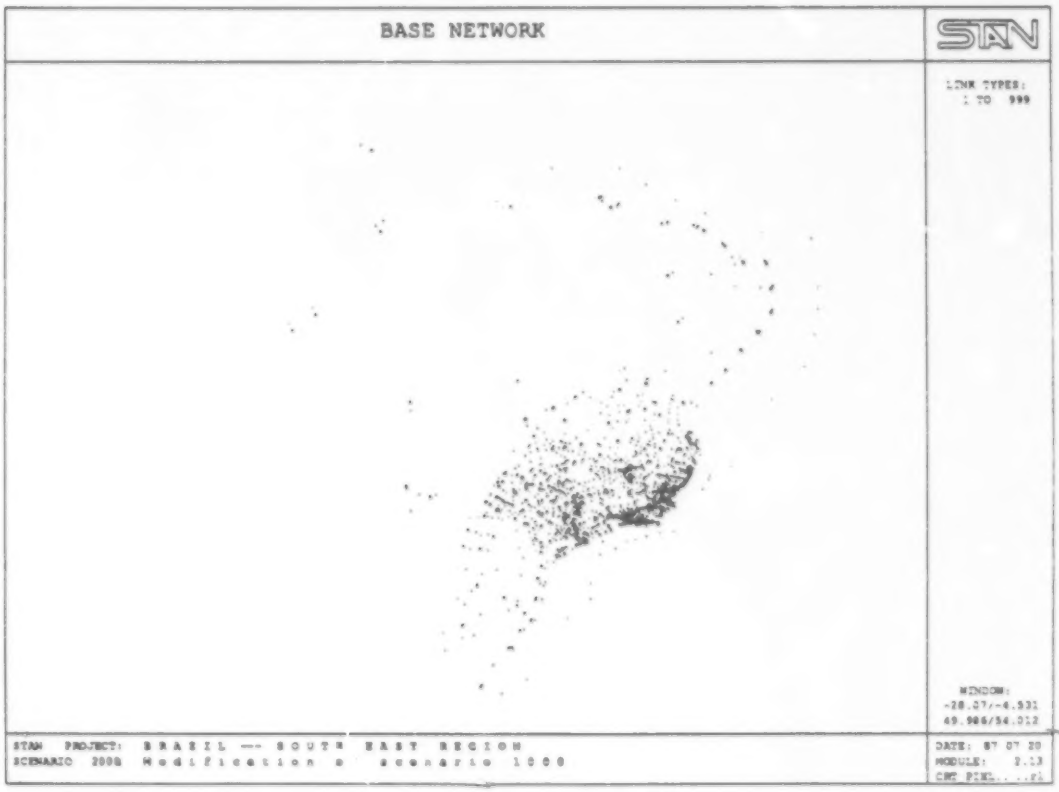
R : to restart the module.

The following example illustrates the use of the **T** command: the selection of five centroids to display the exact distance on shortest path between each of these centroids and the root.



The **W command** allows the user to change the window. This command is explained in detail in section 2.5.3.

The **1,2,...9 commands** are used to superimpose demarcation lines on the plot. The set of demarcation lines that corresponds to the given number is plotted. These commands are only allowed for network plots. The following example shows a plot of the Brazil base network nodes, represented by very small boxes, as it appears initially (without demarcation lines):



The graphic commands 1 3 and 4 are now entered. Since these demarcation sets were defined to contain the borders of Brazil and the states of the South-East region, the resulting plot is:

Note that in general more than one graphic command may be entered on the same line. The **N**, **W** and **S** commands are often used in combination. It is also possible to enter the same command more than once, but usually this is only done with the W command in order to perform complicated window transformations that require more than one window subcommand.

If a command is entered that is not allowed for in the current plot, no error message is displayed, but the bell of the terminal is rung three times to indicate the error.

2.5.3 Windowing

This section describes how to redefine a window once a network plot is completed. To define a window before a plot is carried out, see section 2.4.5.

The user may define the new window by entering the graphic command **W**, which causes the crosshair cursor to appear. Several options are now available to the user: *zooming in*, *zooming out* or *walking around*. These options are described next in detail.

If the user wishes to *zoom in*, that is, to blow up a part of the current window, the crosshair cursor must be positioned to the lower left corner of the part of the window that is to be enlarged, and then the space bar (blank) is depressed. The selected lower left corner is indicated by two lines and the crosshair appear again; the upper right corner is now defined in the same way. The new plot is then displayed.

If the user wishes to *zoom out*, that is, to display more than the contents of the current window, a number between 1 and 9 must be entered by depressing the corresponding key. A new plot is then generated, which is concentric to the current plot but spans an area n (n=1,...,9) times larger than the one displayed on the current plot.

A new center of the window may be established by positioning the crosshair cursor to the desired new center and then typing the **C** key. A new window with the same size as the old window, but shifted to a new enter point is defined in this manner. This operation is also known as *panning*.

The user may also *walk around* on the network plane, that is, display a window of the same size as the current window, that is immediately adjacent to it. This is done by depressing the **U, D, L** or **R** key, which causes the new window to be *up*, *down*, to the *left* or to the *right* of the current window. This option may also be used to produce a sequence of plots that may be glued together to form a big network map.

The coordinates of up to ten windows, that were most recently used in the current module, are stored in a stack. The user may return to the previous window by entering a ^ character.

The current window may be saved for later recall as a predefined window by typing the character >. The crosshair cursor then returns and the user is expected to depress a key from A to Z to indicate which predefined window is to be created. Another window command should then be entered. Windows may only be saved if switch number 5 is set to **on** (see section 2.3.1). Otherwise an attempt to save a window results in three ringings of the bell to indicate that the command has not

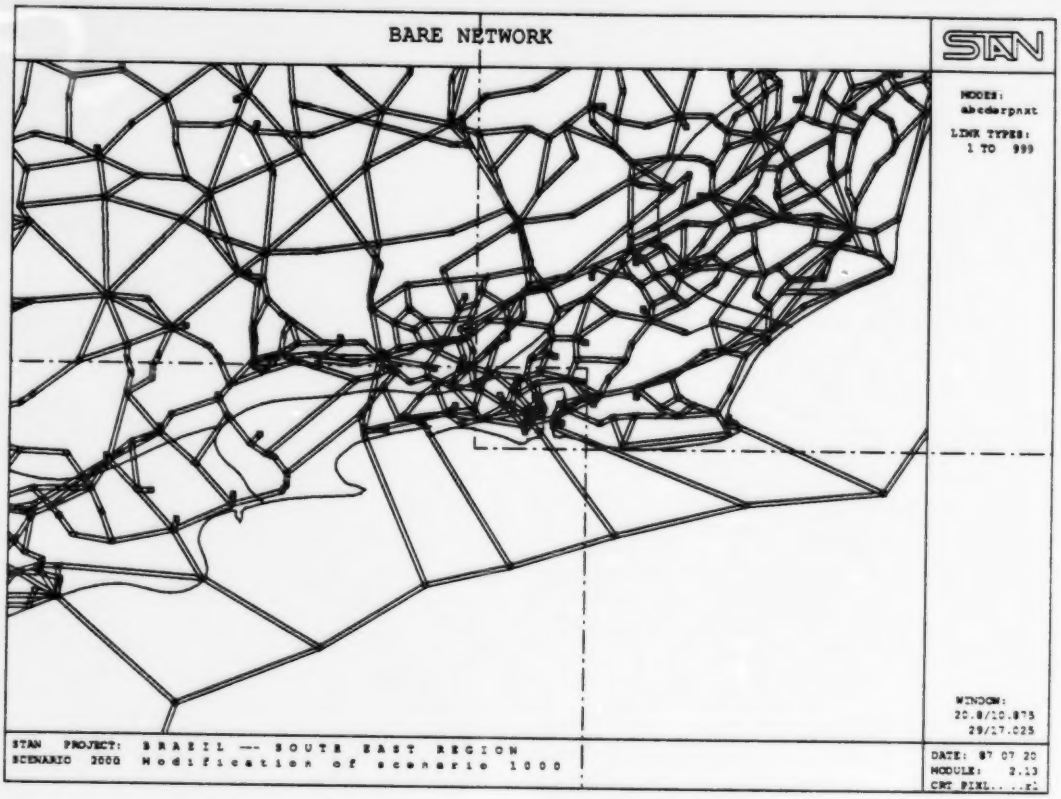
been accepted. Disallowing the saving of windows by setting switch 5 to OFF may be advantageous if, after an initial setup of the predefined windows, the user wants to protects these against unwanted modifications.

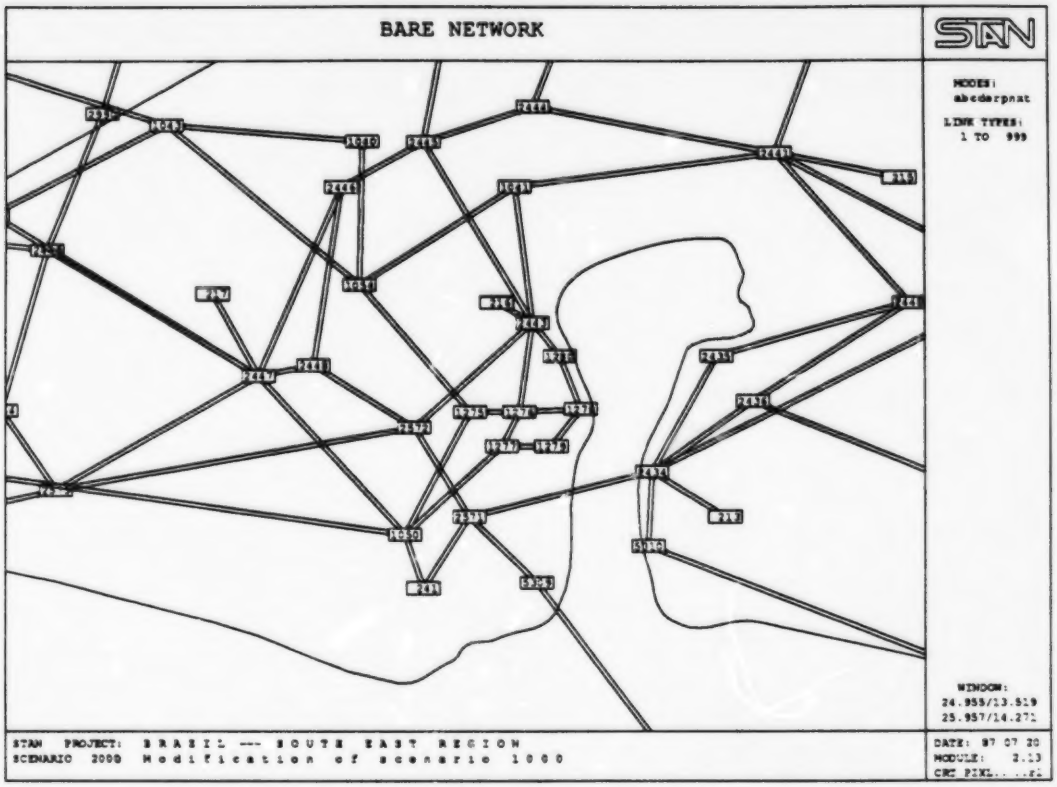
Once a predefined window has been saved using the above sequence, it may be recalled in any subsequent window command by depressing the key < and, upon reappearance of the crosshairs, the corresponding letter A to Z. If an attempt is made to recall a window that has not yet been defined, an error is indicated by three rings of the bell.

Issuing several window commands at the same time may sometimes be used by an experienced user in order to redefine a window in several steps. Consider the following example:

> A user would like to display the window of the same size as the current window that is diagonally adjacent on the upper left corner. Instead of *walking* up using subcommand **U**, replotting, issuing another **W** command, *walking* to the left using subcommand **L** and replotting, the same window could be obtained directly on the next plot by entering two **W** commands on the same line (WW) and responding to the first crosshair with **U** and the second with **L**.

The following example shows the combined effects of the N graphic command and the *zoom in* capability.


BARE NETWORK
STAN
MODES:
abcdefgnxt
LINK TYPES:
1 TO 999
WINDOW:
20.8/10.875
29/17.025
STAN PROJECT: BRAZIL -- SOUTH EAST REGION
SCENARIO 2000 Modification of scenario 1000
DATE: 97 07 20
MODULE: 2.13
CRT PIXL...x1



2.5.4 Additions to plots in manual mode

The graphic command **M** allows the user the access to a *manual mode*, which permits to add manually text and drawings to a completed plot. Once in this manual mode, the user has the option of adding text of different sizes or generating drawings in the form of polygons anywhere on the screen, using the crosshair cursor.

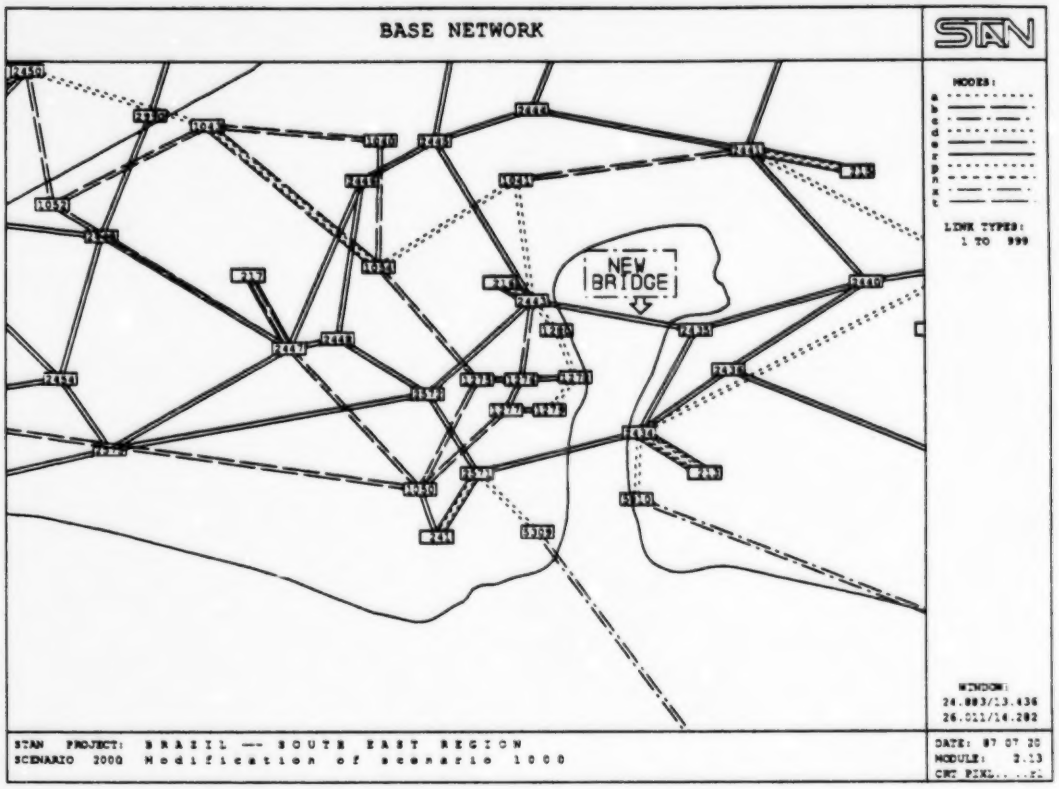
Text may be written by positioning the crosshair cursor to the point where the text should start, and then indicate the desired letter size by depressing one of the keys 1 (big letters), 2, 3 or 4 (smallest letters). The alphanumeric cursor of the

selected size appears then at the desired location and the user may type in a line of text. Each line of text must be terminated by depressing the "carriage return", after which the crosshairs reappear and the user may enter another line of text or a drawing.

A polygon, that is, a sequence of straight lines, may be drawn using one of five different line types: full **(A)**, dotted **(B)**, dot-dashed **(C)**, short dashed **(D)** and long dashed **(E)**. This is done by positioning the crosshair cursor to the starting point, and depressing the appropriate letter (A,B,C,D or E). All the subsequent points are then entered by positioning the crosshair cursor and depressing the space bar (blank). Each time a point is entered the corresponding vector is drawn immediately. The drawing of the current polygon is terminated as soon as a character other than a space (blank) is depressed.

The user may exit the manual mode by typing a **G**. This causes the alphanumeric cursor to appear in the title box at the bottom of the plot. The STAN system is now ready to process another graphic command.

The following example shows the additions made to a plot in manual mode, in order to highlight a network addition with the appropriate annotation.



2.5.5 Cross hatching

The plots generated by the STAN system are based on the drawing of vectors. Graphic objects like boxes, bars and columns are therefore normally just outlined, i.e. the contour of the object is drawn leaving the interior of the object empty. However, the STAN system provides the possibility to hatch (filling by the use of parallel lines) objects that indicate graphically the value of an attribute by a proportional change of their size. Example of such graphic objects are :

- link bars (e.g. link volumes, see module 6.13)
- transfer flows (e.g. transfer volumes, see module 6.13)

- node column (e.g. matrix productions, see module 3.13)

The density of the hatching is defined by a parameter named **hatch width**, normally a module parameter (see section 2.3.3) of the modules in question. The hatch width corresponds to the distance between two neighboring hatching lines defined in hardware coordinate units. A small value therefore results in a darker object than a bigger value. A value of 1 is normally sufficient to make an object appear solidly filled, whereas a value larger than 1280 (diagonal diameter of the plotting surface) will result in an empty object (only contours).

It is to be noted that the execution time of the programs as well as the amount of output generated is inversely proportional to the hatch width. Small values for the hatch width should therefore be used with caution and only when necessary for presentation purposes.

2.5.6 Interrupting graphic output

This section applies only to installations which support user interrupts of the STAN system.

If the user wishes to terminate the drawing of a plot prematurely, this may be achieved by pressing the interrupt key (installation dependent, usually the DELETE or RUBOUT key) while the plot is generated. In this case plotting is stopped, an asterisk (*) is displayed on the upper left corner of the display to indicate that an interruption has occurred and the cursor returns to the graphic command position on the bottom of the screen, where it is ready to accept graphic commands.

It is also possible to interrupt the plotting of demarcations by depressing of the interrupt key.

Note that **no interrupt** should be sent while the **crosshair cursor** is activated, nor when the terminal is in **dialog mode**, since this may cause, on certain systems, an abort of the STAN program.